Edge Deletion Tests and ℓ_1 Regularisation Methods in Graphical modelling for Multivariate Time Series

Rory Ellis
Supervisors: Marco Reale and Chris Price
University of Canterbury
Mathematics and Statistics Department

STAT690

Masters in Statistics

Date of Submission: January 3, 2016

Abstract

In this thesis, the primary aim is to examine graphical modelling in the context of multivariate time series. This work develops on previous work, which provided two approaches, the GMTS and SIN methods, which gave results for the conditional independencies between the variables in datasets. These methods will be compared with a more recent range of methods for estimating the structure of the graphical model, called ℓ_1-regularization, which focused on inducing sparsity in the model. Examining a Gaussian graphical model context, the aim becomes to estimate the covariance/precision matrix, and then produce the partial correlations between variables. This matrix then provides the significant or insignificant edges (lines) between vertices/nodes (variables) in the Conditional Independence Graph (CIG). These methods are compared using a Monte Carlo simulation study, by simulating structural vector autoregressive models (SVAR), which are a mathematical form of representing the dependencies between variables. These simulation studies suggest that the original GMTS and SIN methods produced very useful results in the classification analysis, compared to some of the four ℓ_1-regularization methods used in these studies. Convergence rate analysis and more details on each of these ℓ_1-regularization methods are provided in a comprehensive discussion.

Acknowledgments

A special acknowledgment must be made to Anna Lin, who was kind enough to provide her honours project dissertation and resources to aid with the analysis of this Master's thesis.

Table of Contents

Table of Notation

Symbol	Meaning
n	Number of observations
p	Number of parameters
G	Graph (undirected)
V	Vertex set
E	Edge set
x_i/ x_j/ x_k	Random variables
P	Probability
p	Probability density function
N	Normal Gaussian function
X	Set of random variables
Σ	Covariance matrix
Ω	Precision matrix
Ω_{ij}	Element of Precision Matrix
$\hat{\Sigma}$	Sample covariance matrix
ρ	Partial correlation
β_i	Regression co-efficient
λ	Tuning parameter
r	Order of VAR/SVAR
A_i	Coefficient matrices VAR
t	time
\	Excludes variables in set adjacent
rest	Rest of variables
f_i	functions
ν	Constant vector for VAR
χ^2	Chi- square
$\|\|.\|\|_1$	l-norm
α	Significance level
β	Power of test (in section 5.1)
M	Number of tests
H_o	Null hypothesis
H_a	Alternative hypothesis
μ	Mean
γ(k)	Covariance for x_t and x_{t+k}
C_i	SVAR coefficient matrices
ϵ_t	SVAR error vector
u_t	VAR error terms
T_{Bo}	Bonferroni: unadjusted GMTS test statistics
T_{ua}	Unadjusted GMTS test statistics
λ_L	Lasso tuning paramater
λ_D	Dantzig tuning parameter
M_p	Upper band of l-norm of precision matrices
k	Constant used in comparing convergence rates
s	Number of non-zero elements on the off-diagonal of the precision matrix

MB

ne_a	Neighbourhood of variable a
b	Rest of variables
θ	Vector coefficients
$\hat{\theta}^{a,\lambda}$	Lasso estimates of θ_a

SPACE

ϵ_i	Error of x_i
$\Omega^{ij}/\Omega^{ii}/\Omega^{jj}$	Elements of precision matrix
ρ^{ij}	partial correlation between x_i/x_j
Θ	upper triangular partial correlations
w_i	non-negative weights of the i^{th} regression
λ	tuning parameter
$\|\|.\|\|_F$	Frobenius norm

Glasso

$\hat{\Sigma}$	sample covariance matrix
W	estimated covariance matrix
λ	tuning parameter
I	identity matrix
$\hat{\beta}$	estimated regression coefficient

GMTS

t	critical value
ν	residual degree of freedom from regression
$\hat{\rho}_{l,j}$	estimated partial correlation between x_l and x_j
$\hat{\tau}_{ij}$	critical value from equation

CLIME

$\hat{\Sigma}$	sample covariance matrix
λ	tuning parameters
$\hat{\Omega}_{ij}$	elements of estimated precision matrices
e_i	standard unit vector in RP

TIGER

N_p	p dimension norm distribution
$\hat{R}$	sample correlation matrix
ζ	constant tuning parameters
\i	all variables excluding i

SIN

S	Significant set
I	Intermediate set
N	Non-significant set
$\hat{G}_S$	Graph from significant set of edges
$\hat{G}_{SI}$	Graph from S U I

(1) Introduction

The aim of this thesis is to examine the use of graphical modelling in a multivariate time series settings. This work develops on a previous study (Lin, 2008) which compared two approaches for testing the presence of edges in the graph. These two tests were a t-test proposed by Reale and Wilson (2001) and SIN, a test introduced by Drton and Perlman (2008). The methods which are compared with these original two, glasso (Friedman et al, 2007), SPACE (Peng et al, 2009), CLIME (Cai et al, 2011), and TIGER (Liu & Wang, 2012), are all ℓ_1-regularized methods which are designed to induce sparsity in the graphical models.

Graphical modelling provides a useful form of analysis for multivariate time series (MTS), because it can be used to identify dependencies between variables. A vector autoregressive (VAR) model is a good representation for MTS, since correlations can be determined from the equations, and linear dependencies can be found from lagged variables to contemporaneous variables, as desirable. Issues with understanding relationships between contemporaneous variables become difficult with this model, so MTS are often fitted using a structural VAR model (SVAR)

Having a parsimonious model like the SVAR model is important when sparsity is required in the graphical model. Sparse graphs have relatively small number of edges, and make interpretation simple (Friedman et al, 2008). In most cases, the edges which should be omitted from the graph are unknown, so it is necessary to use the data to make this choice. In recent years, the ℓ_1 (lasso) regularization approach has been considered by many authors for this purpose. The four methods listed before use this regularization and their performances will be compared in this thesis.

There have been many applications for graphical models examined over the years. Originally this approach was examined in analyzing paths for genetic problems analysis, and then moved into areas like social sciences and economics. There have also been applications in gene regulatory networks (Gottard & Pacillo, 2010) and speech recognition. Climate research is also an important area to consider when considering graphical modelling, because it is possible to understand all the dependencies between variables. In fact, analysis of a climate dataset will be provided in this research.

The premise of graphical modelling is as follows. Given a set of random variables, the aim is to represent the relationships between variables (or lack of relationships). A node

in the graph refers to a random variable or vector in the dataset. If A and B are conditionally independent, then this is shown as

$$A \perp B$$

Which indicates no line (a line is known as an edge in graphical modelling) between the two nodes A and B. Graphical modelling provides a nice platform for representing these relationships, as well as tools for model selection of these models. In fact, examining the Gaussian graphical model context, model selection is possible by identifying the link between the inverse of the covariance matrix (precision matrix), and the partial correlations. There are many caveats associated with model selection in graphical modelling. The multiple testing of hypotheses for each of the partial correlations provides issues with type 1 and 2 errors, as well as multicollinearity, which must be amended in order to carry out analysis.

The main motivation of this thesis, however, was to examine ℓ_1-regularization methods in estimating these graphical models. Now, more than ever, high dimensional data – data which has many more random variables than the number of observations, i.e. $p \gg n$ – has become more popular to fit graphical models to. Normally, this would be an issue, because the empirical covariance matrix would become singular. However, these newly introduced methods are designed to overcome this issue, and still provide a sparse structure for the graphical model.

Simulation studies are derived from the previous work in this area, to compare the old methods used, and the newest ones introduced in this thesis. This was done to determine which method could estimate the model the best, in terms of determining the correctly including or excluded edges from the graphical model.
Because these newer methods are convex optimization problems, another measure of performance that must be considered is the computation time required to carry out the processes. As a result, convergence rates of each of the methods are discovered, and compared with one another, to determine which method converged to the solution fastest.

Now it is possible to state how the thesis will proceed. Section 2 examines the implementation of Conditional Independence Graphs (CIGs) for multivariate time series, and includes the model which will be used to represent these graphs. Motivation for considering ℓ_1-regularization will be provided here too, as well as a short examination of convex optimization. Section 3 provides a literature review on some of

the approaches that have been proposed in edge deletion tests and ℓ_1-regularization in graphical modelling. Section 4 then considers the methods that were not only used in the analysis for this thesis, but methods that were used previously in this field. Section 5 quickly looks at some primary issues that need to be taken under consideration when running the tests. Section 6 provides the results from the simulation studies that were conducted to compare the results to be introduced. Section 7 documents some issues and discussions that were found from the comparisons and from the papers read. Section 8 provides some insight on future research that can be done in this area, including potential improvements on the studies conducted here. Finally Section 9 provides some remarks on this research.

(2) Introducing the Problem

In this thesis, we consider a graph, which consists of a list of vertices (nodes), and a set of edges which connect pairs of vertices. In the context of graphical models, a vertex represents a random variable, and it provides a way of understanding the joint distribution of the entire set of random variables which is being studied. The case that will be examined in this thesis will be that of an undirected graph (also known as a Markov random field or a Markov network). In this kind of graph, the exclusion of an edge corresponds to conditional independence between two variables, given the rest of the variables in the set. These concepts will be defined more in detail in the next subsections.

(2.1) Graphical model:

The Conditional Independence Graph (CIG), which is a graphical model G = (V, E), with undirected graph, G, must first be considered.

- The set of vertices, V, corresponds to the random variables in the model
- The Edge set, E, shows the conditional (in)dependence relations between variables.

In order to understand the idea of the conditional independence relations between variables, it may be useful to simplify the problem and consider conditional independence in the context of simple events. In this scenario, conditional independence of events is defined as follows:
Events A and B are conditionally independent given a third event C if

$$P(A \cap B|C) = P(A|C)P(B|C)$$

This may also be written as:

$$P(A|B \cap C) = P(A|C)$$

This concept can now be considered for random variables. If three random variables x_i, x_j, and x_k are random variables or random vectors, with a density $p(x_i, x_j, x_k)$. Then x_i is conditionally independent of x_j given a third variable x_k, (written $x_i \perp x_j \mid x_k$) if:

$$p(x_i, x_j|x_k) = p(x_i|x_k)p(x_j|x_k)$$

which, again, can be written as

$$p(x_i|x_j, x_k) = p(x_i|x_k)$$

If, for example, x_i is in fact conditionally dependent on x_j, given the rest of the variables/vectors in the set, then there will be an edge between the two nodes in the graphical model, and this makes them "adjacent". This can be shown as $x_i \sim x_j$.

This can finally be defined for a conditional independence graph. Now consider a set of random variables $X = (x_1,\ldots,x_p)$. Here each of the random variables corresponds to a variable of a dataset. These variables follow a joint distribution p (the joint distribution that will be used in the studies in this thesis will be described later). In the case of a Markov graph G, not having an edge between two variables implies conditional independence between them, given the rest of the variables. This can be shown by the following notation

$$x_i \perp x_j \,|X_{\setminus\{xi,\, xj\}}$$

Where $\setminus\{x_i,x_j\}$ indicates the variables in the set X excluding x_i and x_j. This property is associated with the *pairwise Markov independencies* of G.

It is also useful to consider the global Markov property. For this, the idea of separation has to be considered. Considering three subgraphs, A, B, and C, of the graph G, then C separates A and B if every path between A and B intersects a node in C. This separator C gives a nice property, in that is breaks the graph into conditionally independent pieces. Defining this, if C separates A and B, then $A \perp B|C$. This is the global Markov property of G

Using this information, it is possible to provide an example of an undirected graph. Imagine that there are five variables in a dataset, denoted a, b, c, d, and e. The links between variables can be represented as an undirected graph as displayed in Figure 2.1. These links can also be represented in a matrix form, referred to as an adjacency matrix, where a 1 in the matrix refers to a significant dependence between two variables, and a 0 refers to conditional independency. This relates to the property that was raised previously, where an edge between two random variables means that they are "adjacent". The adjacency matrix of the example from Figure 2.1 is also shown in Table 2.1.

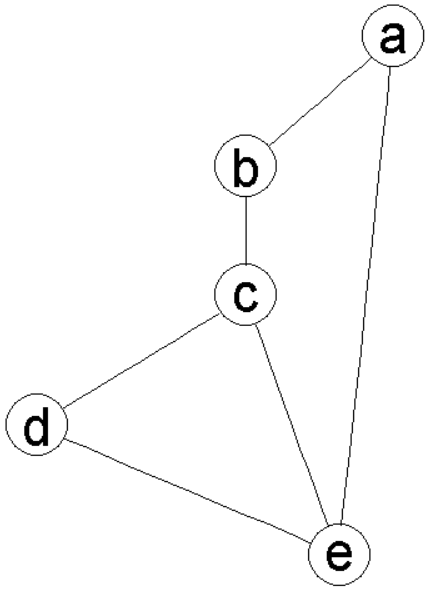


Figure 2.1: Example of an undirected graph for 5 variables

	a	b	c	d	e
a	0	1	0	0	1
b	1	0	1	0	0
c	0	1	0	1	1
d	0	0	1	0	1
e	1	0	1	1	0

Table 2.1: Adjacency matrix showing dependencies between the variables from Figure 2.1

(2.1.1) Gaussian Graphical Model:

Now it is possible to examine the context which will be considered in this thesis. The Gaussian distribution provides a useful distribution to implement in terms of graphical models, due to the convenience of producing partial correlations between the variables.

Assuming that the collection of random variables, $X = (x_1,\ldots,x_p)$ follows a multivariate normal distribution, i.e. $X \sim N(\mu,\Sigma)$, with mean μ and positive definite matrix Σ, then the distribution's probability density function is:

$$f(x;\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^p|\Omega|}} exp(-\frac{1}{2}(x-\mu)^T\Omega(x-\mu)) \qquad (2.1)$$

Where, as stated above, $\Omega = \Sigma^{-1}$.

The covariance matrix of the data, called the sample covariance matrix, $\hat{\Sigma}$, is given by:

$$\hat{\Sigma} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)\,(x_i - \mu)^T \tag{2.2}$$

(2.1.2) Precision Matrix

The precision matrix can be used in recovering the sparse structure of a graph, as well as estimating the graphical model itself. This is because a 0 (or significantly small) value in the precision matrix corresponds to a conditional independence between the two variables. Also, another reason behind this is that it contains a link to the partial correlation matrix, through the simple transformation:

$$\rho_{i,j} = -\frac{\Omega_{ij}}{\sqrt{\Omega_{ii}\Omega_{jj}}} \tag{2.3}$$

Where:

- ρ_{ij} represents the i,jth element of the partial correlation,
- Ω_{ii}, Ω_{jj} , and Ω_{ij} represent elements of the precision matrix Ω,

This transformation can be achieved by identifying that the partial correlation coefficient, denoted $\rho_{i,j}$, between x_i and x_j is given by $\rho_{i,j} = x_i \perp x_j | X_{\setminus\{xi,\, xj\}}$. Therefore, when considering two variables x_i and x_j that are conditionally independent, then

$$x_i \perp x_j | x_k \leftrightarrow \rho_{ij|rest} = 0 \tag{2.4}$$

where rest refers to the rest of the variables in the dataset.

(2.1.3) Marginal Distribution

The marginal distribution for the multivariate normal distribution is simply the probability distribution of a subset of the random variables in a dataset. For example, given the variables a, b, c, d, and e, then the marginal distribution of a and e is simply a multivariate normal distribution with the mean vector $\mu' = (\mu_a, \mu_e)$ and covariance matrix

$$\Sigma' = \begin{pmatrix} \Sigma_{a,a} & \Sigma_{a,e} \\ \Sigma_{e,a} & \Sigma_{e,e} \end{pmatrix} \tag{2.5}$$

(2.1.4) Conditional Distribution

The conditional distribution of one variable versus the rest is also useful to define. Using the approach from Hastie et al (2008), partition X = (Z,Y), where Z = ($x_1,\ldots,x_{p-1}$)

consists of the first p-1 variables, and Y = x_p is the last variable. The conditional distribution of Y given Z is then given by:

$$Y|Z = z \sim N(\mu_Y(z - \mu_Z)^T \Sigma_{ZZ}^{-1}\sigma_{ZY}, \sigma_{YY} - \sigma_{ZY}^T \Sigma_{ZZ}^{-1}\sigma_{ZY}) \quad (2.6)$$

Where the covariance matrix, Σ, has been partitioned by

$$\Sigma = \begin{pmatrix} \Sigma_{ZZ} & \sigma_{ZY} \\ \sigma_{ZY}^T & \sigma_{YY} \end{pmatrix} \quad (2.7)$$

(2.1.5) ℓ_1-regularization

Issues arise in the inversion of the covariance matrix, Σ, when the sample size, n, is smaller than the number of parameters, p. In particular p ≫ n, the empirical covariance Σ is singular, so it is not possible to access information about the conditional independencies, let alone invert the matrix (Banerjee et al, 2006). These issues will be addressed by some methods examined in this thesis.

There has been a variety of research carried out to estimate the structure of a graph in a graphical model. When it is unknown what edges of the graph to omit, the data must be used to discover the conditional independencies. This has led to the idea of the ℓ_1-regularization. In the methods section these approaches will be described individually in more detail, but an introduction to ℓ_1-regularization will be provided here.

Meinhausen and Bühlmann (2006) take a simple approach to the ℓ_1-regularization approach that was first referred to in Section 1, where rather than trying to fully estimate the covariance matrix Σ or the precision matrix Ω, the aim is to only estimate which of the components Ω_{ij} of the precision matrix are nonzero. In order to achieve this, a variation of lasso regression was fitted using each variable as the response, and the rest as the predictors. The lasso can be shown as:

$$\hat{\beta}(\lambda) = \arg\min_{\beta} \left(\frac{\|Y - X\beta\|_2^2}{n} + \lambda \|\beta\|_1 \right) \quad (2.8)$$

where $\|Y - X\beta\|_2^2 = \sum_{i=1}^{n}(Y_i - (X\beta)_i)^2$ is the residual sum of squares, $\|\beta\|_1 = \sum_{j=1}^{p}|\beta_j|$, and $\lambda \geq 0$ is a penalty parameter. The lasso estimator has the effect of shrinking certain coefficients $\hat{\beta}_j(\lambda)$ to zero for a specified λ, hence giving λ the term "shrinkage" parameter.

The scenario under the Meinhausen and Bühlmann approach for the lasso will be described in the methods section.

A more systematic approach was considered, by first observing that the log-likelihood of the data can be found by:

$$\ell(\Omega) = log\ det(\Omega) - trace(\hat{\Sigma}\Omega) \tag{2.9}$$

The idea is then to maximize the penalized log-likelihood via

$$log\ det(\Omega) - trace\left(\hat{\Sigma}\Omega\right) - \lambda\|\Omega\|_1 \tag{2.10}$$

where $\|\Omega\|_1$ is the sum of the absolute values of the elements of Ω. This should not be confused with the actual 1-norm of the precision matrix, which would be the maximum absolute column sum of the matrix. This is the component that induces sparsity on the maximum likelihood, because it has the effect of reducing down the coefficient values to zero. The negative of this penalized likelihood is a convex function of Ω, which leads to the convex optimization problem to be described later in the section. This approach was considered by Banerjee et al, in 2007, and then continued by Friedman et al in 2008; leading to the method called the graphical lasso, or glasso, and will be described in more detail in the methods section.

The SPACE method (Peng et al, 2009) aims to penalized the joint loss function with the penalty based on the precision matrix, instead of the precision matrix**.** The CLIME method (Cai et al, 2011), meanwhile aims to minimize the precision matrix (i.e. find the sparsest matrix), using the ℓ_1-norm, within a feasible set of maximum likelihood estimates. Finally, the TIGER method (Liu and Wang, 2012) uses a variation of the Lasso, the SQRT-Lasso (Belloni et al, 2012), and uses a ℓ_1-penalty based on each column of the precision matrix. More details on these methods will be provided in Sections 3 and 4

(2.1.6) Sparsity

The question remains, however: Why is sparsity required in graphical modelling? The first reason provided is that it provides a better interpretation of the data. When applying the methods used in this thesis to real-world examples, it seems unnecessary and unrealistic to have every variable in a dataset conditionally dependent on all the others, simply due to the fact that this does not give much insight into what the people implementing the analysis are trying to figure out about the data.

An advantage of considering sparsity in a method is that it provides the opportunity to improve on the computational efficiency of the estimations of the graphical model. As will be discovered in the proceeding sections of the thesis, some methods, due to the

nature in which they estimate the partial correlation matrices, require a lot of computational time to estimate the elements.

Inducing sparsity also reduces the risk of overfitting in the model. The ideas of stationarity and multicollinearity will be discussed in detail later in the thesis. But simply put, reducing the inherent dependency between the variables (and also on the previous time points) helps in providing a better estimate of the graphical model. If a simple model explains the data well enough (up to a significant point), then it is unnecessary to introduce more edges into the model.

(2.1.7) Convex Optimization:

Because we are looking at some aspects of optimizing (using the ℓ_1-penalty), it is necessary to describe convex optimization. According to Boyd & Vandenberghe (2004) a convex optimization problem can be observed as the form:

$$\begin{aligned} &\text{Minimize } f_o(x) \\ &\text{subject to } f_i(x) \leqslant b_i,\ i=1,\ldots,m, \end{aligned} \tag{2.11}$$

where the functions $f_o, \ldots, f_m: \mathbb{R}^n \rightarrow \mathbb{R}$ are all convex, so they satisfy:

$$f_i(\alpha x + \beta y) \leqslant \alpha f_i(x) + \beta f_i(y) \tag{2.12}$$

for all $x, y \in \mathbb{R}^n$, $\alpha, \beta \in \mathbb{R}$ with $\alpha + \beta = 1$, $\alpha \geq 0$, $\beta \geq 0$.

These types of problems are very similar to least squares or linear programming problems. In fact, these two approaches are considered special cases of the more general convex optimization problem described above.

Providing an optimization problem that is convex provides a few key advantages, through the theory that is involved in the procedure.

1. If a local minimum exists, then it is a global minimum.
2. The set of all (global) minima is convex.
3. For each *strictly* convex function, if the function has a minimum, then the minimum is unique.

Particularly the first and third points here are important to remember, when trying to find the minimum to the functions associated with each of the methods that will be examined in this thesis.

(2.2) VAR and SVAR models:

Vector autoregressive (VAR) models were first introduced into statistical analysis in the early 1990's with prime examples by Lütkepohl (1993). But, in fact, these models were considered well before that time when Sims (1980) advocated the model's use over the other models available at the time. This reasoning was attributed to having too many restrictions based off a priori knowledge for the other large macroeconomic models around in that time. The VAR model was first implemented to discover the underlying characteristics of time series. These characteristics include studying the dynamic interactions between variables, impulse analysis, and forecast error variance decompositions. Particularly in the context of this thesis, it was also possible to identify the correlations between variables, since the coefficients of the model showed the linear dependencies between the random variables.

Unfortunately, while this type of model provides a flexible and general framework to undertake these analyses, a VAR model has caveats. When an unrestricted VAR model is considered, then each variable is expressed as a function of its own lagged variables and all the other variables in the system, meaning that many parameters have to be estimated for even a moderate number of random variables.

Another issue becomes apparent when the model is defined, because it is not possible, using the VAR model, to consider the contemporaneous linear dependencies between variables (i.e. in the same time point). Therefore a structural VAR (SVAR in some publications) can be used because, as will be shown later in the section, there are a few advantages to using this model.

However in both of these models, the risk of overfitting still remains. Therefore model reduction must be implemented to create a sparser model. This section provides the VAR and structural VAR framework, and methods will be provided later in the thesis, to provide ways of finding sparse structures of these models. After this, it is then possible to change the implementation of these methods into a graphical model, which is a way of identifying the dependence relations between variables in a time series framework, which can be represented using a structural VAR model.

(2.2.1) Vector Autoregressive (VAR) models

A VAR(r) model (i.e. a VAR model of order r), can be represented as:

$$y_t = \nu + A_1 y_{t-1} + \cdots + A_r y_{t-r} + u_t, \quad t = 0, \pm 1, \pm 2, \qquad (2.13)$$

Where

- $y_t = (y_{1t}, \ldots, y_{pt})'$ is a (px1) random vector;
- A_i are fixed (pxp) coefficient matrices, and
- $\nu = (\nu_1, \ldots, \nu_p)'$ is a fixed (px1) vector of intercept terms, which accounts for the possibility of $E(y_t) \neq 0$.

There is also $u_t = (u_{1t}, \ldots, u_{pt})$, which is a (px1) vector of error terms. This vector satisfies:

- $E(u_t) = 0$;
- $E(u_t u_t') = \Sigma_u$, and
- $E(u_t u_s') = 0$ for $s \neq t$

An example of how a VAR model looks is to use a VAR(1) model which is represented in matrix notation as:

$$\begin{pmatrix} y_{1,t} \\ y_{2,t} \end{pmatrix} = \begin{pmatrix} \nu_1 \\ \nu_2 \end{pmatrix} + \begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{pmatrix} \begin{pmatrix} y_{1,t-1} \\ y_{2,t-1} \end{pmatrix} + \begin{pmatrix} u_{1,t} \\ u_{2,t} \end{pmatrix} \qquad (2.14)$$

This matrix equation can be turned into a system of two equations as:

$$y_{1,t} = \nu_1 + A_{1,1} y_{1,t-1} + A_{1,2} y_{2,t-1} + u_{1,t} \qquad (2.15)$$

$$y_{2,t} = \nu_2 + A_{2,1} y_{1,t-1} + A_{2.2} y_{2,t-1} + u_{2,t} \qquad (2.16)$$

The coefficients $A_{i,j}$ are used to show the linear dependence between the variables. For example, the coefficient $A_{1,1}$ shows the linear dependence of $y_{1,t}$ on $y_{1,t-1}$ in the presence of $y_{2,t-1}$ (Tsay, 2014). In other words, the current observations (at time t) of each component depend on lagged variables of its own or other time series.

In order to estimate a VAR model, firstly the order of the VAR model must be determined, then the parameter selection. For the VAR model order selection, there are a few methods available to use, some of which are provided by Lütkepohl (2005). These include the impact that the fitted VAR order has on the forecast MSE. The Likelihood Ratio Test Statistic can also be implemented, as well as a testing scheme for VAR order determination. However the method that will be examined in this research is the use of

Information Criterion, including the Akaike Information Criterion (AIC), or the Bayesian Information Criterion (BIC).

(2.2.2) Structural Vector Autoregressive (SVAR) models

As was stated in the introduction to this section, the structural VAR (SVAR) model is a more suitable framework when identifying graphical models, due to the model's ability to show the contemporaneous conditional (in)dependencies between variables. In fact, the VAR(r) model examined above is a reduced form model, whereas the SVAR is intuitively a structural form model.

The structural (VAR) SVAR model can be represented as:

$$C_0 y_t = \Phi + C_1 y_{t-1} + \cdots + C_p y_{t-p} + \varepsilon_t \tag{2.17}$$

There are a few differences between the VAR and the SVAR models. Firstly, there is the C_0 coefficient matrix associated with the y_t variable, i.e. the instantaneous effects. C_o is shown as;

$$C_0 = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ c_{2,1} & 1 & & 0 \\ \vdots & & \ddots & \vdots \\ c_{p,1} & c_{p,2} & \cdots & 1 \end{pmatrix} \tag{2.18}$$

which is a lower-triangular matrix. It must be noted that this C_o matrix is only possible when considering Conditional Independence Graphs (CIGs). The reason behind this is that it is possible to have directed dependencies, such as those observed in Directed Acyclic Graphs (DAGs). This matrix implies that there is no directed dependency between 2 variables. Therefore it is possible to consider conditional independencies between the variables in the current time point, instead of only the lagged variables, which provides a more realistic interpretation in graphical modelling

In this case $C_i = C_o A_i$, $\Phi = C_o \nu$, and $\varepsilon_t = C_o u_t$,

$$E(\varepsilon_t \varepsilon_t') = \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix} = \Sigma_a \tag{2.19}$$

What this suggests is that the off-diagonal elements of Σ_a are zero, so the error terms are independent. This characteristic is what makes the VAR model a structural representation.

If there are no zeros in the coefficient matrices (C_o (lower diagonal), C_1, …, C_p), then the SVAR model is considered saturated. However, it is often found that the lagged

variables do not have a significant level of dependency on explaining the current variables, so the SVAR becomes sparse. In fact, it is usually expected that the SVAR model is more parsimonious (sparse) than the corresponding VAR model. Sparse SVAR structures can be obtained from a Directed Acyclic Graph (Wilson et al, 2001). However, central to the construction of the DAG is the CIG, and hence why this thesis focusses on finding the later

(2.3) Conditional Independence Graphs (CIG)

Conditional independence graphs are a very useful platform to implement when the aim is to find relationships between variables in Multivariate Time Series. There is also the opportunity to transform a CIG into a DAG. For this Graph, the variables are represented as vertices and the pairwise conditional independencies are shown as edges. For a CIG, the only dependence that solely occurs from one variable to the other (and not vice versa) is the dependence from vertices associated with previous time points to more current time points. Otherwise, all edges are symmetric, i.e. an edge from x_i to x_j implies dependence both ways.

Expanding on Section 2.3.1, A CIG is constructed by computing the pairwise partial correlations between variables, and then using model selection to determine the significant conditional dependencies.

(2.3.1) Constructing a CIG

Suggested by its name, the CIG is formed by considering the conditional independencies between variables. Here, an omitted edge between two vertices suggests that the two associated variables are conditionally independent. Therefore it can be stated that if there is no edge between two variables x_i and x_j, then they are independent given the rest of the nodes.

Applying the knowledge from Section 2.1, it is possible to use the relationship between the precision matrix and the partial correlation matrix in deriving the eventual CIG. Only three steps are required to provide the Conditional Independence Graph from a dataset:

1. Compute the partial correlation matrix showing the strength of relationships between variables.
2. Use a thresholding or filtering procedure to test for the significant conditional dependencies, thus creating an adjacency matrix.

3. Relate this newly formed adjacency matrix from the MTS data to a graph.

Examining step 2 in more detail, the adjacency matrix, as stated previously in Section 2.1, consists of zeroes and ones. Placing this idea in the context of a CIG, this can be defined as:

$$\begin{cases} 1 \; if \; \hat{\rho}_{ij} \geqslant threshold \\ 0 \; if \; \hat{\rho}_{ij} < threshold \end{cases}$$

where a 1 ultimately leads to an edge between two nodes in the CIG, and a 0 means no edge.

(3) Literature Survey

It is not possible to provide a literature survey of an array of studies that have been conducted in the field of graphical modelling. It should be noted that this can be seen as a "preliminary" literature survey, because the ℓ_1-regularization methods that are implemented in this research bear differing levels of resemblances to other forms of the types of ℓ_1-regularization methods available. Therefore a secondary literature survey will be provided later in this thesis, with more detailed descriptions of the methods involved

The idea of using the precision matrix to estimate a graphical model first arose when Dempster, in 1972, proposed the idea of covariance selection. In this thesis, the motivation was to simplify the structure of the covariance matrix, by setting elements of the precision matrix to zero. A simple forward selection approach was used, with a Newton-type algorithm implemented after estimating the correlation matrix in each iteration. To determine which edges should be included in the graphical model, a crude t-statistic was used. Vichik and Oshman (2011) observe that this optimization method is in fact not optimal in most minimum mean square error (MMSE). The method proposed is found to satisfy the optimality conditions considered in the thesis, whereas Dempster's approach does not.

Drton and Perlman (2008) identify a method SIN) that, instead of testing each correlation in the partial correlation individually, it tests them all simultaneously. This is achieved by using Fisher's z-transform, and Šídák's inequality to provide p-values based off the partial correlation matrices, which could then be compared with the significance level α, to identify the edges in the graphical model. Interestingly, this method provides a third set, called the indeterminate set I, which identifies edges which may or may not be significant depending on the significance level chosen. Gottard and Pacillo (2010) improve on the SINful approach, by examining the issues the SIN method has with outliers. The minimum covariance determinant (MCD) provides a significant improvement to this approach, over the SIN method.

Meinhausen and Bühlmann, in 2006, pioneered the use of the lasso regression in determining the sparse structure of a graphical model. This was done by using a neighbourhood selection approach, which estimated the conditional independence restrictions separately for each node in the graph. This made the approach equivalent to variable selection in Gaussian linear models. The penalty parameter associated with the

lasso approach used in this method was determined by requiring a constraint on the probability of falsely connecting two distinct components of the true graph. In this case, the neighbourhood pursuit method (denoted MB) proved to be useful with $p > 30$ parameters, because previous greedy forward-backward approaches (Edwards, 2000), became computationally inefficient. In fact, the MB approach worked for $p > 1000$ nodes (parameters).

Yuan and Lin (2007) propose an alternative to the standard 1-norm penalty used in the MB approach (and other methods proposed later), considering a non-negative garrotte-type estimator. A BIC-criterion was used as a selection approach for the penalty parameter in this case. In comparisons with the MB and SIN approaches, the measurement used to compare performances, called the Kullback-Leibler loss (KL), stated that this approach was competitive with the other 2 approaches. It was discovered that the SIN approach in particular provided a poor false negative rate performance, whereas this nonnegative garrotte-type estimator has issues with false positive rates.

Verzelen & Villers (2009) consider the work described in the previous two parts, and identify a different procedure in the hypothesis testing framework. The focus becomes on comparing a minimal graph (all edges are significant) with graphs containing missing edges (in the context of gene regulation networks). This testing is implemented in a high-dimensional context, where the number of parameters, p, is greater than the sample size, n. Neighbourhood analysis, like the approach carried out by Meinhausen and Bühlmann, is then used to determine the edges to be included in the graphical model. It is warned that the weights in this method correspond to a Bonferroni choice of weights. Therefore, if the number of variables, p, is large, then the testing procedure may suffer from a loss of its size.

El Ghaoui et al (2006) discuss what happens when the sample size, n, is much smaller than the number of parameters, p, stating that the empirical covariance matrix, $\hat{\Sigma}$, becomes singular, so information cannot be accessed regarding the conditional independencies between variables.
Two methods, Nesterov's method, and a block coordinate descent method, are used to find the sparse covariance matrix, using a ℓ_1-norm penalized maximum likelihood. Nesterov's method has the effect of producing a complexity estimate of the problem, while the block coordinate descent approach has the property of promising a positive-definite matrix that, as will be discussed later, is an important characteristic. It is found

that the block coordinate descent method in particular provides useful and interpretable results, with a very high number of variables (p=1000 in the case examined).

Banerjee et al, in 2007, identify that there are a few caveats associated with the MB approach, and attempt to amend these issues in their paper. In most situations, the estimated covariance matrix and the sample covariance matrix are not equal, which is assumed in the MB approach. This means that it does not provide the maximum likelihood estimate. In order to achieve a maximum likelihood estimate for this approach, the dual of the penalized log-likelihood is derived, and two approaches are used to determine the structure of the graphical models. The first is a block coordinate descent method, which is an optimization approach for solving the dual, and the second approach considers Nesterov's first order method to solve the original penalized log-likelihood. A tuning parameter is derived for this approach using an adaptation of the student t-distribution. It is found that these two approaches work well for datasets with thousands of variables, with reasonable computational efficiency.

The paper by D'Aspremont et al (2008) follows a similar path to the approach considered by Banerjee et al, proposing that the dual of the ℓ_1-penalized log-likelihood should be found, and optimization used on this equation. The aim here is to find a sparse representation of the sample dataset used, in order to show conditional independencies between the variables. Again, a block coordinate descent method is used on the dual problem, as well as a smooth optimization approach. In the case of the Smoothed optimization method, it was found that there was a trade-off between better dependence on the problem size, and a worse dependence on accuracy.

Friedman et al (2008) use the work conducted by Banerjee et al (2007) as inspiration to consider the graphical lasso (glasso), which is used to estimate the sparse structure of the inverse covariance matrix. The block coordinate descent method is adopted and modified to create a more computationally efficient algorithm (30-4000 times faster than the method by Banerjee et al.).

Rothman et al (2008) propose a lasso-type estimator called "Sparse Permutation Invariant Covariance Estimation" (SPICE), which is found to be exactly the same estimator as the one used by Yuan and Lin (2007) described previously. The main aim of this thesis is to observe the convergence rates of the approach. An iterative algorithm is developed for computing the SPICE-estimator using Cholesky decomposition. The advantage over other methods in this area of analysis is that the method is invariant

under permutations of the variables, hence where the name of the method comes from. Cross-validation is used to determine the tuning parameter, which provides the level of sparsity on the model.

Peng et al (2009) identify a method called Spatial PArtial Correlation Estimation (SPACE, with spelling intended), which is designed to improve on the work done by Meinhausen and Bühlmann (2006). One of the main motivations of this approach was to provide a method which used the symmetric property of the covariance/precision matrices, to increase the computational efficiency of the MB approach. Again, it is a neighbourhood approach, design for high dimension, low sample size settings, which identifies the sparsity in a neighbourhood (or subset/cluster) in the matrix. More improvements over the MB approach, and even the glasso approach, will be discussed later in the methods section.
An active shooting algorithm is also implemented to solve the lasso regression problems more efficiently, by updating the coordinates iteratively until convergence occurs. The tuning parameter used in this method is determined by using a BIC criterion. The partial correlations are estimated by a penalized loss function, and focus on individual regressions (like the Meinhausen and Bühlmann approach). However, it also simultaneously performs neighbourhood selection for all variables.

Another method of ℓ_1-penalization is considered to improve on previous methods. Cai et al (2011) consider another approach of estimating the sparse inverse covariance matrix, called the CLIME method (Constrained ℓ_1 Inverse Matrix Estimation). The aim in this scenario, however, is to minimize the 1-norm of the precision matrix subject to a constraint to be defined later. It is then found that this problem can be decomposed into p vector minimization problems.
This method mainly focusses on high-dimensional cases, where p is greater than n. The CLIME method is compared with the refitted CLIME, which accounts for biases introduced by the ℓ_1-penalty, the glasso method, and the Smoothly-fitted Absolute Deviation (SCAD) penalty by Fan et al. It was found that the Refitted CLIME in particular performed better than the other 2 methods on sensitivity (True Positive Rate), and provides similar results on specificity (True Negative Rate). The other classification performance measure, Matthew's correlation coefficient (MCC), suggests a 25% improvement over the other methods. The CLIME method also provides the sparsest matrix, which helps in terms of interpretability.

Hsieh et al (2011) consider a novel approach to the optimization problem using the ℓ_1-penalized Gaussian maximum likelihood. This is done by using Newton's method, and applying quadratic approximation. Iterative coordinate descent is then used to solve the resulting lasso problem.

An Armijo-based rule is considered, in order to obtain the step-size that not only ensures a sufficient descent to the solution, but also positive definiteness in the inverse covariance matrix.

The data is split into free and fixed sets, to determine which variables should be updated, using the stationary condition of the Gaussian Maximum Likelihood Estimate, where using Newton's update on the fixed set will not change the results. This leads to the possibility of implementing block coordinate descent, resulting in improved efficiency.

Liu and Wang (2012) propose the TIGER method (Tuning Insensitive Graph Estimation and Recovery), which is designed to be more computationally efficient than previous methods, with extensive comparisons with the CLIME method mentioned above undertaken. The method introduced for estimating high dimensional Gaussian graphical has a tuning-insensitive property, which means that the optimal regularization parameter selection for λ does not depend on any unknown parameters. Like CLIME, this is a method that adopts a column-by-column regression scheme. However, one feature that discerns the TIGER method from other methods is its use of the SQRT-Lasso, proposed by Belloni et al (2012). There are a few advantages which are alluded to throughout the thesis, the first being that this approach is tuning-free, so the whole dataset can be used to select the model from the data, instead of using cross-validation/subsampling methods used in other methods, which is said to make the TIGER method more computationally cheap. Also, it is computationally simple, and can be scaled for large datasets.

In comparisons with the CLIME and glasso methods, there were attempts made to determine which method produced the best results, using false positive and negative rates (FPR and FNR). Based off the ROC curves produced from the simulation analysis, it was concluded that the TIGER method performs better than the CLIME and glasso methods in the high dimensional settings, indicating that it could adapt better to inhomogeneous noise models. The TIGER method also performed better in terms of the regularization parameter, as well as the negative log-likelihood estimates.

Cai et al (2012) consider using the Adaptive CLIME (ACLIME) which develops on the work conducted by Cai et al (2011) on the CLIME method. The advantage over the CLIME method in this case is that the tuning parameter, λ, changes depending on the column of the data, whereas for the CLIME method, it is universal for the model, although both parameters are determined by Cross Validation.

In the analyses conducted in this thesis, the SIN, CLIME, SPACE, glasso, and TIGER methods are used, as well as a GMTS approach considered in previous work in this area. Therefore, more details and discussions will be provided for these approaches later in the thesis.

(4) Methods

Now it is possible to discuss the methods that have been used in the past to produce graphical models, along with the methods that will be considered in this research. The first approach in section 4.1 will be stepwise selection, which was introduced in the context of graphical modelling by Edwards in 2000. In 4.2 and 4.3, the GMTS and SIN methods will be described respectively. In 4.4, the neighbourhood selection introduced by Meinhausen and Bühlmann will be shown, as well as some caveats associated with the process. Finally, in sections 4.5 to 4.8, the four ℓ_1-regularization methods will be considered, firstly with SPACE, then Glasso, then CLIME, and finally with TIGER.

(4.1) Forward/Backward Stepwise Selection

Details on this method are provided by Edwards, 2000. This method involves incrementally searching for the edges that are significant enough to be included in the final graphical model. Using an initial model, the edges in the model are either added or removed at each iterate, until some criterion which is used to determine the best model is satisfied. At each of the steps, deciding which method should be included or removed is determined by a significance test. In the case of Edwards, a χ^2- test is implemented to determine the significant edges in the model. The two stepwise approaches examined in detail are the main methods in this line of work, the forward and backward stepwise procedures.

(4.1.1) Forward Stepwise Selection

In this method, a simple model, which is usually considered inconsistent with the data, is started with, and edges are added onto the model at each iterate, given that the edge is both significant, and improves the model enough given a certain criterion. For a situation where sparsity is wanted, this method is preferred, firstly because there will be fewer steps required to get to the best estimate, and secondly because there will be fewer issues regarding the existence of the maximum likelihood estimate (since we are only using a simple model here).

(4.1.2) Backward Stepwise Selection

In this case, the full model is considered, where all edges are included in the model. Edges are removed if their corresponding p-values are higher than the significance level, with the edge of highest p-value eliminated first, as long as it improves the model

enough. The other caveat is making sure that there is still enough information in the model so that the results can be interpreted properly.

Comparing the forward and backward methods, the backward approach starts with a model which, while it is more complex, is initially consistent with the data. On the other hand, the forward stepwise approach starts with a null model which is most likely inconsistent with the data. Therefore the backward approach seems more suitable in this respect.

(4.2) GMTS

First introduced in 2001, Wilson et al consider the GMTS approach, which considers the link between the entries of the partial correlation matrix and the critical t-value, to determine which entries of the partial correlation matrix should be equal to zero. A threshold based on a critical value

$$critical\ value = \frac{t}{\sqrt{t^2+\nu}} \tag{4.1}$$

was defined in this case. In this equation, the t stands for the critical value, and ν stands for the residual degrees of freedom in the regression (Wilson & Reale, 2008). The null hypothesis H_o is then rejected if $|\hat{\rho}_{i,j}| > crit$

- $|\hat{\rho}_{i,j}| \leq crit$ implies a conditional independence between the two variables x_i and x_j. This means that there will be no edge in the conditional independence graph between the two varibles.
- $|\hat{\rho}_{i,j}| \geq crit$ implies a significant partial correlation between the two variables at the given significance level. Therefore there will be an edge between the two variables in the CIG.

(4.3) SIN

Drton and Perlman (2008) consider an alternative approach to finding the structure of the graphical model. This approach focusses on the Gaussian context, appling Fisher's z-transformation, Šídák's correction inequality, and Holm's step-down procedure, to test the multiple hypotheses simultaneously, where the hypotheses can be stated as:

$$H_o: \hat{\rho}_{i,j} = 0 \quad \text{vs} \quad H_a: \hat{\rho}_{i,j} \neq 0$$

The null hypothesis in this case determines whether the associated variables x_i and x_j are conditionally independent, stating whether an edge should not be included in the graphical model. The approach provides p-values which are compared with the significance level provided to determine which edges are significant in the final model. For this method, the significant set of edges is denoted *S*, and the non-significant set edges are called *N*.

An interesting aspect of this approach is that not only is there a set of conditional dependencies and independencies between the variables in the dataset, but there is a third set of variables, called an intermediate set *I*, which identifies the edges which may be significant under a different significance level. This provides an interesting analysis, because there is the opportunity to examine the effect of choosing different significance levels on the structure of the graphical model. While it is not examined in this thesis, a smaller model $\hat{G}_S$, whose edges correspond to the p-values in the significant set S, can be compared with the larger model $\hat{G}_{SI}$, whose edges correspond to the p-values from $S \cup I$.

(4.4) Meinhausen and Bühlmann (2006)

In order to improve on some of the caveats involved in the selection procedure discussed above, Meinhausen and Bühlmann considered a neighbourhood selection approach to identify the sparse structure of the graphical model. A neighbourhood selection procedure is introduced and implemented by the authors, where the method is considered a "subproblem" of covariance selection. In the paper, a series of lasso regressions, which would identify the zeroes in the precision matrix, was proposed. First, the approach must be examined in more detail

For graphical model estimation, the authors used this approach in neighbourhood selection. In this method, "a neighbourhood ne_a of a node $a \in V$ is the smallest subset of $V \setminus \{a\}$ so that, given all variables $x(ne_a)$ in the neighbourhood, x_a is conditionally independent of all remaining variables". The neighbourhood of a node $a \in V$ consists of all nodes $b \in V\setminus\{a\}$, such that $(a,b) \in E$, given that a and b are a pair of random variables. Therefore, given n i.i.d. observations of X, this approach aims at individually estimating the neighbourhood of any given variable. This problem can be considered a standard regression problem.

In this method, it should be noted that the number of nodes in the graph and the distribution generally depend on the sample size, so V = V(n) (the vertices) and Σ = Σ(n) (the covariance matrix). As stated before, when attempting to predict the variable X_a from all the other variables $\{X_i ; i \in V(n) \setminus \{a\}\}$, the Lasso coefficient estimates have the effect of asymptotically identifying the neighbourhood of a node a in the graph, as will be shown.

The lasso estimate $\hat{\theta}^{a,\lambda}$ of θ^a is given by

$$\hat{\theta}^{a,\lambda} = arg \min_{\theta:\theta_a=0} \left(\frac{1}{n}\|x_a - X\theta\|_2^2 + \lambda\|\theta\|_1\right) \quad (4.2)$$

Where θ^a is a vector of coefficient used for optimizing prediction

$$\theta^a = arg \min_{\theta:\theta_a=0} E\left(x_a - \sum_{i\in V(n)} \theta_k X_k\right)^2 \quad (4.3)$$

Therefore, $\|\theta\|_1 = \sum_{b\in V(n)}|\theta_b|$is the ℓ_1-norm of the coefficient vector.

When considering an individual regression, the $\lambda\|\beta\|_1$ induces sparsity on the estimates of the regression coefficients, given that the regularization parameter λ is large enough. This results in a variable selection procedure, which is attractive in terms of the motivation behind this thesis.

The assumptions made in this approach establish a few key considerations that are required in the estimation of the graphical model.

- The first assumption looks at high dimensionality, stating that the number of variables, p, is able to grow, as the sample size n is raised to an "arbitrarily high power".
- The next assumption assures that an empirical variance can be achieved, by scaling the variables appropriately, which is important in this situation, in order to lose the dependence on the chosen units or dimensions from which they are represented.
- There is also a limit on the rate of growth of the neighbourhood, to keep a level of sparsity in the final solution.

(4.4.1) Issues

There are a few caveats that come with implementing this method:

- As stated by Peng et al (2009), in this neighbourhood selection approach, sparsity is only imposed on the neighbourhoods, which becomes an issue if sparsity needs to be considered for the whole partial correlation matrix.
- This method does not consider the symmetric nature of the partial correlation matrix (refer back to the example provided for the undirected graph, where a link from a to b implies a link from b to a). This means that the method estimates $(p-1)^2$ parameters, while some methods, such as the SPACE method which is introduced next, only estimates $p(p+1)/2$ parameters, thus improving computational efficiency.
- It is possible that the neighbourhood pursuit approach does not provide sign consistency, for example in the lasso regressions for this approach, it may occur that the sign of the estimated regression coefficient $\hat{\beta}_{ij}$ is different from the sign of $\hat{\beta}_{ji}$.
- Banerjee et al (2007) provides some insight on the Meinhausen and Bühlmann approach as well, stating that the first major difference between their approach (the one leading to the glasso approach which is about to be introduced) and the neighbourhood pursuit approach is that each penalized regression problem has a unique solution, due to the regularization that is undertaken. Also, the problem data is updated after each regression, so it can be seen as a recursive lasso.

(4.5) SPACE

The first method examined is one that was considered by Peng, Wang, Zhou, & Zhu (2008), which considered a symmetric regression approach called "Sparse PArtial Correlation Estimation" (SPACE). This approach was created due to the lack of symmetry in the Meinhausen and Bühlmann method. The idea was to fit the model:

$$x_i = \sum_{i \neq j} \beta_{ij} x_j + \varepsilon_i \tag{4.4}$$

Where x_i and x_j are random variables, ε_i is the corresponding disturbance term, and β_{ij} is the population regression coefficient of y_i on y_j (Friedman et al, 2010):

$$\beta_{ij} = \rho_{ij} \sqrt{\frac{\Omega_{jj}}{\Omega_{ii}}} \tag{4.5}$$

Like above, ρ_{ij} is the partial correlation between x_i and x_j, and Ω_{ii}/Ω_{jj} are the elements of the precision matrix. Note here how the partial correlation and precision matrix

elements i, j, are identified using superscripts instead of subscripts. The reason behind this will become apparent soon. Based off this information, a penalized joint loss function was considered:

$$L_p(\theta, \sigma, X) = \frac{1}{2}\left(\sum_{i=1}^{p} w_i - \left\|X_i - \sum_{j\neq i}\beta_{ij}x_j\right\|^2\right) + \lambda\|\Theta\|_1 \quad (4.6)$$

$$= \frac{1}{2}\left(\sum_{i=1}^{p} w_i - \left\|X_i - \sum_{j\neq i}\rho^{ij}\sqrt{\frac{\Omega^{jj}}{\Omega^{ii}}}X_j\right\|^2\right) + \lambda\|\Theta\|_1 \quad (4.7)$$

Where w_i are nonnegative weights for the regressions, and the second term denotes the ℓ_1-penalty (like a lasso regression), where

$$\lambda\|\Theta\|_1 = \lambda\sum_{1\leqslant i<j\leqslant p}|\rho_{ij}| \quad (4.8)$$

In other words, $\Theta = (\rho_{12}, \ldots, \rho_{(p-1)p})^T$ An Active-Shooting algorithm, inspired by previous work, is proposed to create a computationally efficient algorithm for solving lasso regressions like the one above. The aim is to minimize the ℓ_1-penalized loss function, but the algorithm also alternates between estimating the Ω_{ii} and the ρ_{ij}. Friedman et al (2010), state that this method minimizes the function

$$\frac{1}{2}\|X - XB\|_F^2 + \lambda\sum_{i\neq j}|\rho_{ij}| \quad (4.9)$$

i.e. the penalized Frobenius norm. The Frobenius norm is defined by the square root of the sum of absolute squares of the elements of the matrix in question.

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2} \quad (4.10)$$

For this method, it is important to be able to estimate a precise tuning parameter λ for the method. Because it is a lasso regression, there are a variety of methods that can be used to select the parameter. However, in this case, a simple and computationally easy approach is implemented, called a "BIC-type" criterion, where BIC stands for Bayesian Information Criterion. First of all, the residual sum of squares for the i^{th} regression is found for the SPACE estimator

$$RSS_i(\lambda) = \sum_{k=1}^{n}\left(y_i^k - \sum_{j\neq i}\hat{\rho}_{ij}^{\lambda}\sqrt{\frac{\Omega_{jj}^{\lambda}}{\hat{\Omega}_{ii}^{\lambda}}}\, y_j^k\right)^2 \quad (4.11)$$

The BIC-type criterion can then be defined as:

$$BIC_i(\lambda) = n \times \log\big(RSS_i(\lambda)\big) + \log(n) \times \#\{j: j \neq i, \hat{\rho}_{ij}^{\lambda} \neq 0\} \qquad (4.12)$$

This is where it becomes important to have the superscript for the elements of the matrices, to provide the opportunity to place the λ in the equation (4.11), for the residual sum of squares.

The final part to consider in this method is the choice of weights used in the process (w_i in the penalized joint loss function above). Wang et al provide three different options to choose for the weights:

1. Uniform weights, where $w_i = 1$;
2. Residual variance-based weights, where $w_i = \hat{\sigma}_{ii}$;
3. A degree based weight, where w_i is proportional to the estimated degree of y_i, i.e. $\#\{j : \hat{\rho}_{ij} \neq 0, j \neq i\}$

For the first iteration for each model, the initial weight is set to be one, then new weights are calculated based on the conditions for each of the weight choices.

(4.6) Glasso

Referring to work carried out by Banerjee et al, in 2007, Friedman et al works on the ideas introduced by Meinhausen and Bühlmann. Firstly, the aim is to maximize the penalized log-likelihood

$$log(det(\Omega)) - tr\big(\hat{\Sigma}\Omega\big) - \lambda\|\Omega\|_1 \qquad (4.13)$$

Where tr(.) refers to the trace, and the $\|\Omega\|_1$ refers to the ℓ_1-norm, i.e. the sum of the absolute values of the elements of the inverse covariance matrix. Here, λ is used as a tuning parameter, like in the neighbourhood pursuit approach considered by Meinhausen and Bühlmann.

From previous work it was shown that this problem is convex, and the estimation of the covariance matrix Σ is considered rather than its inverse Σ^{-1}. A block coordinate descent method can be utilized by optimizing over each row and subsequent column of W, where W is the glasso estimate of Σ. W and $\hat{\Sigma}$ can be partitioned by:

$$W = \begin{pmatrix} W_{11} & w_{12} \\ w_{12}^T & w_{22} \end{pmatrix} \qquad \hat{\Sigma} = \begin{pmatrix} \hat{\Sigma}_{11} & \hat{\Sigma}_{12} \\ \hat{\Sigma}_{12}^T & \hat{\Sigma}_{22} \end{pmatrix} \qquad (4.14)$$

Then, it is discovered that the solution to w_{12} satisfies a box-constrained quadratic problem (QP), which is able to be solved by an interior-point procedure. This is because w_{12} satisfies:

$$w_{12} = argmin_y \left\{ y^T W_{11}^{-1} y \middle\| y - \hat{\Sigma}_{12} \right\|_\infty \leqslant \rho \right\} \qquad (4.15)$$

(4.6.1) Duality

Banerjee et al also examine the convex dual, to show that solving (4.15) is the same as solving the dual problem

$$min_\beta \left\{ \frac{1}{2} \left\| W_{11}^{1/2} \beta - b \right\|^2 + \lambda \|\beta\|_1 \right\} \qquad (4.16)$$

Where $b = W_{11}^{1/2} \hat{\Sigma}_{12}$. It can be seen in this equation (4.16) that it is similar to a lasso approach.

The difference between the neighbourhood pursuit approach (MB), and the graphical lasso method is that $W_{11} \neq \hat{\Sigma}_{11}$ in general, so the MB approach does not give the maximum likelihood estimator. While in previous work, it has been pointed out that the blockwise interior point method considered here is equivalent to solving and updating the lasso problem above, only in this thesis has it been implemented, because the fast coordinate descent methods make getting to the lasso problem faster, and therefore more desirable.

The glasso algorithm can be described as so:

1. Starting with $W = \hat{\Sigma} + \rho I$, the diagonal of W remains unchanged in the proceeding steps.
2. For $j = 1, 2, \ldots, p$, solve the lasso problem (4.16) described above, taking the input of the inner products W_{11} and s_{12}. This provides a p-1 vector solution $\hat{\beta}$. Then the corresponding row and column of W is filled in using $w_{12} = W_{11}\hat{\beta}$.
3. Continue algorithm until convergence occurs.

(4.7) CLIME

Cai et al (2011) propose another method of estimating a sparse precision matrix, called the CLIME method (Constrained ℓ_1 Inverse Matrix Estimation). For this method, the CLIME estimator is defined as so: Let $\{\hat{\Omega}_1\}$ be the solution to the optimization problem

$$\min\|\Omega\|_1 \text{ Subject to:}$$

$$\left\|\hat{\Sigma}\Omega - I\right\|_\infty \leq \lambda, \qquad \Omega \in \mathbb{R}^{p\times p} \tag{4.17}$$

Where λ is the tuning parameter, $\hat{\Sigma}$ is the sample covariance matrix, and I is the identity matrix.

In this method, the symmetry condition on the precision matrix Ω is not used, so the solution to (1) is not symmetric. As a result, the CLIME estimator of Ω is found by symmetrizing $\widehat{\Omega}_1$ using the following approach.

Denote $\widehat{\Omega}_1 = \left(\widehat{\omega}_{\mathrm{ij}}^1\right) = (\widehat{\omega}_1^1, \dots, \widehat{\omega}_{\mathrm{p}}^1)$. Mathematically, the final CLIME estimator can be defined as:

$$\widehat{\Omega}_1 = \left(\widehat{\Omega}_{ij}\right), \text{where}$$

$$\widehat{\Omega}_{ij} = \widehat{\Omega}_{ji} = \widehat{\Omega}_{\mathrm{ij}}^1 I\{|\widehat{\Omega}_{\mathrm{ij}}^1| \leqslant |\Omega_{\mathrm{ji}}^1|\} + \Omega_{\mathrm{ji}}^1 I\{|\widehat{\Omega}_{\mathrm{ij}}^1| > |\widehat{\Omega}_{\mathrm{ji}}^1|\}. \tag{4.18}$$

Putting this equation into words, it simply means that between the $\widehat{\Omega}_{\mathrm{ij}}^1$ and $\widehat{\Omega}_{\mathrm{ji}}^1$ elements, the one with the smallest absolute magnitude is used.

Relating to the methods examined before, the CLIME estimator introduced in (4.17) can be further decomposed into p vector minimization problems. Consider e_i to be a standard unit vector in $\mathbb{R}^{\mathrm{p}}$, with 1 in the i^{th} coordinate, and 0 otherwise. For $1 \leqslant i \leqslant p$, $\hat{\beta}_i$ is the solution to the convex optimization problem:

$$\min\|\beta\|_1 \text{ subject to}$$

$$\left\|\hat{\Sigma}\beta - e_i\right\|_\infty \leq \lambda_n \tag{4.19}$$

Where β is a vector in $\mathbb{R}^{\mathrm{p}}$.

It must be noted that another thresholding step must be taken to recover the graph. In detail, define a threshold estimator $\widetilde{\Omega} = \widetilde{\Omega}_{ij}$ with

$$\widetilde{\Omega}_{ij} = \widehat{\Omega}_{ij} I\{|\widehat{\Omega}_{ij}| \geq \tau\} \tag{4.20}$$

where $\tau \geq 4M_p\lambda$ is the new tuning parameter, with λ the tuning parameter from estimating the 1-norm of the precision matrix, and M_p is the upper bound of the 1-norm of the precision matrix Ω. It is then possible to define:

$$M_p(\tilde{\Omega}) = \{sgn(\tilde{\Omega}_{ij}), 1 \leqslant i,j \leqslant p\}$$
$$M_p(\Omega_0) = \{sgn(\Omega_{ij}^0), 1 \leqslant i,j \leqslant p\}$$
$$S(\Omega_0) = \{(i,j): \Omega_{ij}^0 \neq 0\}$$

where S is the support of Ω_o, where Ω_o is the precision matrix.

An explanation of the Dantzig selector will be provided in Section 7.

(4.7.1) Link to glasso

The paper that introduces this method also contains information on the link between the CLIME method and the graphical lasso method introduced earlier by Friedman et al (2008). In order to carry out this comparsion, a condition, introduced by Ravikumar et al (2008) must be considered.

Irrepresentable Condition: There exists some $\alpha \in (0,1]$ such that

$$\left\|\Gamma_{S^cS}(\Gamma_{SS})^{-1}\right\|_{\ell_1} \leq 1 - \alpha$$

Where $\Gamma = \hat{\Sigma}^{-1} \otimes \hat{\Sigma}^{-1}$, S is the support of Ω, and $S^C = \{1,\ldots,p\} \times \{1,\ldots,p\} - S$.

This is quite a strong assumption that the estimates of the zero elements of the precision matrix Ω are exactly zero with high probability. Xue and Zou (2012) state that the CLIME method contains nice theoretical properties, without having to adhere to this strong assumption.

(4.8) TIGER

The final method examined is the TIGER method (Tuning-Insensitive Graph Estimation and Recovery), examined by Liu and Wang in 2012. The main difference between this method and other methods, which is considered an advantage according to the authors, is that it contains an asymptotic tuning-free property. This allows the user of the method to use the entire dataset to run analysis.

Prior to examining the procedure, it may be useful to consider column-by-column regression which the CLIME also relies on in its calculations, as well as some methods that will be examined in detail in the discussion section

Firstly, a reminder on the conditional distribution is necessary. Given a set of random variables $X \sim N_p(0,\Sigma)$, where p symbolizes a p-dimension normal distribution, the conditional distribution of x_j given $x_{\setminus j}$ (\j means the rest of the variables):

$$x_i | x_{\setminus i} \sim N_{p-1}\left(\Sigma_{\setminus i,i}\left(\Sigma_{\setminus i,\setminus i}\right)^{-1} x_{\setminus i}, \Sigma_{ii} - \Sigma_{\setminus i,i}\left(\Sigma_{\setminus i,\setminus i}\right)^{-1}\Sigma_{\setminus i,i}\right) \quad (4.21)$$

Denote $\alpha_i = \left(\Sigma_{\setminus i,\setminus i}\right)^{-1}\Sigma_{\setminus i,i} \in \mathbb{R}^{p-1}$, and $\sigma_i^2 = \Sigma_{ii} - \Sigma_{\setminus i,i}\left(\Sigma_{\setminus i,\setminus i}\right)^{-1}\Sigma_{\setminus i,i}$. Then

$$x_i = \alpha_i^T x_{\setminus i} + \varepsilon_i \quad (4.22)$$

where $\varepsilon_i \sim N(0, \sigma_i^2)$ is independent of $x_{\setminus i}$. Using a block matrix inversion formula:

$$\Omega_{ii} = \left(Var(\varepsilon_i)\right)^{-1} = \sigma_i^{-2}, \quad (4.23)$$

$$\Omega_{\setminus i,i} = -\left(Var(\varepsilon_i)\right)^{-1}\alpha_i = \sigma_i^{-2}\alpha_i \quad (4.24)$$

Therefore, the precision matrix Ω can be recovered in a column-by-column fashion by regressing x_j on $x_{\setminus j}$ for j = 1, 2, …, p. Then it is possible to denote $\alpha_i := (\alpha_{i1},\ldots,\alpha_{i(p-1)})^T \in \mathbb{R}^{p-1}$. Various adaptations of this approach have been used for this approach, like the MB approach, and by Yuan (2010) using the Dantzig selector. Sun and Zhang (2012) consider this approach using the scaled-lasso, which has a few similarities to the TIGER method. Cai et al (2011) use this type of regression too, as well as Liu and Luo, who build on the work done on the CLIME estimator.

The reason that this method is considered "tuning-insensitive" is because of the use of the SQRT-Lasso, which was introduced by Belloni et al in 2012, which was used to estimate both the graph G, and the precision matrix Ω simultaneously.
For a linear regression problem y = Xβ + ε, the SQRT-Lasso estimates the coefficients β by solving the following equation:

$$\hat{\beta} = \arg\min\left\{\frac{1}{\sqrt{n}}\|y - X\beta\|_2 + \lambda\|\beta\|_1\right\} \quad (4.25)$$

Unlike the lasso approach, it was shown in the original author's paper that choosing the penalty parameter λ does not depend on any unknown parameters in the model.

Now to describe the approach used to estimate the graph and precision matrix. First consider $\hat{\Gamma} := diag\left(\hat{\Sigma}\right)$ to be a p-dimensional diagonal matrix, where the diagonal elements of the matrix are the same as those in $\hat{\Sigma}$. Now:

$$Z \coloneqq (Z_1, \dots, Z_p)^T = X\hat{\Gamma}^{-\frac{1}{2}} \tag{4.26}$$

Using the definition of x_i above:

$$Z_i\hat{\Gamma}_{ii}^{1/2} = \alpha_i^T\hat{\Gamma}_{\setminus i,\setminus i}^{1/2}Z_{/i} + \varepsilon_i \tag{4.27}$$

two new variables have to be introduced

$$\beta_i \coloneqq \hat{\Gamma}_{\setminus i,\setminus i}^{\frac{1}{2}}\hat{\Gamma}_{i,i}^{-\frac{1}{2}}\alpha_i \quad and \quad \sigma_i^2\hat{\Gamma}_{i,i}^{-1} \tag{4.28}$$

Therefore Z_i can be redefined as

$$Z_i = \beta_i^T Z_{\setminus i} + \hat{\Gamma}_{ii}^{-\frac{1}{2}}\varepsilon_i \tag{4.29}$$

Finally, the sample correlation matrix $\hat{R}$ can be defined as

$$\hat{R} \coloneqq \left(diag(\hat{\Sigma})\right)^{-1/2}\hat{\Sigma}\left(diag(\hat{\Sigma})\right)^{-1/2} \tag{4.30}$$

and the precision matrix estimator can be provided now. For j = 1,…,p, the j^{th} column of Ω is estimated by:

$$\hat{\beta}_i \coloneqq \arg\min_{\beta_i \in \mathbb{R}^{p-1}} \left\{\sqrt{1 - 2\beta_i^T\hat{R}_{\setminus i,i} + \beta_i^T\hat{R}_{\setminus i,\setminus i}\beta_i} + \lambda\|\beta_i\|_1\right\} \tag{4.31}$$

$$\hat{\tau}_i \coloneqq \sqrt{1 - 2\hat{\beta}_i^T\hat{R}_{\setminus i,i} + \hat{\beta}_i^T\hat{R}_{\setminus i,\setminus i}\hat{\beta}_i} \tag{4.32}$$

$$\hat{\Omega}_{ii} = \hat{\tau}_i^{-2}\hat{\Gamma}_{ii}^{-1} \quad and \quad \hat{\Omega}_{\setminus i,i} = \hat{\tau}_i^{-2}\hat{\Gamma}_{ii}^{-\frac{1}{2}}\hat{\Gamma}_{\setminus i,\setminus i}^{-\frac{1}{2}}\hat{\beta}_i \tag{4.33}$$

In the paper, Liu and Wang state that choosing the tuning parameter λ to be defined as

$$\lambda \coloneqq \zeta\pi\sqrt{\frac{\log p}{2n}} \tag{4.34}$$

where ζ ranges between [√2/π, 1]. This gives optimal rates of convergence in the asymptotic setting. A discussion of the tuning parameter choices for each method will be provided later.

(5) Model Selection Issues:

As a starting point in the testing, it may be useful to discuss some considerations that need to be taken when implementing these tests. These aspects may affect the authenticity of the results that have been obtained through the simulation studies in particular, so it is important that proper examination of the data is undertaken prior to analysis. In the next subsections, the idea of type 1 and type 2 errors are considered, where a type 1 error represents including an edge between x_i and x_j when it shouldn't be in the graphical model, and a type 2 error is failing to include an edge in the model when it is in the present in the true graph.

(5.1) Multiple Testing

The first concept examined is that of multiple testing. Consider a situation where there is a significant conditional dependence between x_i and x_j, referring back to the hypotheses of the tests involved, this suggests that the H_o is false (should be rejected) and the H_a is "true". On the other hand, if there is no significant dependence, then H_o is true, and H_a is false. When it comes to implementing this approach on the sample partial correlation matrix, the graphical model is determined by selecting the edges between x_i and x_j based on which null hypotheses H_o have been rejected. The significance level α is used to determine the type 1 error rate (for the analyses carried out it is set at 0.05). This means that the probability of incorrectly finding an edge between x_i and x_j is 5%. However, when multiple hypothesis tests are carried out, the "combined" type 1 error rate becomes significantly larger than 5%.

Consider a scenario where the null hypothesis is true for each of the tests conducted, and each of the tests is independent. The probability of finding an incorrect link is then found by $1 - (1 - \alpha)^M$, where M is the number of tests conducted.

In this situation, there are two common definitions of the combined type 1 error rate. These are the Family Wise Error Rate (FWER) and the False Discovery Rate (FDR). The FWER is the probability that when all the null hypotheses are true, at least one null hypothesis will be rejected. The FDR is the expected proportion of all null hypotheses that are incorrectly rejected.

Conclusion	H_o: no edge, is true	H_a: there is an edge, is true
No edge between x_i and x_j	Correct decision $p = 1 - \alpha$; A	Type 2 error $p = \beta$; C
Edge between x_i and x_j	Type 1 error $p = \alpha$; B	Correct decision $p = 1 - \beta$; D

Table 5.1: Table for FWER and FDR

If we use table 5.1 above as an example, it can be seen that the FWER is the probability that B is at least 1 among the (A+B) null hypothesis that are true. The FDR is the expected value of B/(B+D), which is equal to 0 if (B+D) = 0.

To account for this, an FWER control is used by using the Bonferroni adjustment, where in the case of the simulation studies, the significance level for the individual tests is α/M. There is a caveat with this approach though, because this adjustment results in a loss of statistical power for the test, i.e. the probability of rejecting the null hypothesis given that the null hypothesis is false (excluding a link when the link is present in the true graph). As shown in the table, the power is given by $1-\beta$, where β is the type 2 error rate.

This provides a trade-off between the type 1 errors and type 2 errors, where allowing a higher probability for the type 1 error results in a low probability of type 2 errors and vice versa. In this case, it is up to the person implementing the methods to decide which error is more dangerous in terms of the results given. However, an obvious way of amending this issue is to increase the sample size, which has been done in the simulation studies.

(5.2) Multicollinearity

The next issue is the idea of multicollinearity, which is a necessity to examine when modelling time series. This concept arises when two or more variables are highly dependent on each other. In certain cases, it is inevitable that a degree of multicollinearity will arise when dealing with time series, because the time series x_t may be highly correlated with its lagged variables (x_{t-1} etc.) or the contemporaneous variables. In the case of graphical modelling, this can become a dangerous issue, because if one variable is linearly dependent on other variables, the coefficient estimates (which will determine the dependency between the variables) can change erratically with small perturbations.

This exacerbates the issues discussed in the previous subsection, particularly with the adjustment procedures affecting the statistical power. This is because the methods may incidentally adjust the significance levels for the individual tests by more than what is required in order to control the FWER, resulting in further deteriorating the statistical power of the test.

(5.3) Stationarity

The final issue to consider is that of stationarity. In the previous subsection, it was stated that there will be some inherent dependency between the variable x_t and its lagged variables. This is expected, and in fact is accounted for in the simulation studies because, as will be shown, the models from which process is simulated allows for lagged dependencies. However, at the same time, it is important that this "time dependence" is not a long term one, least of all because the further back the dependency lasts for the variables, the more edges that will have to be tested between variables, creating a more computationally inefficient procedure.

The conditions for stationarity are as follows:

(5.3.1) Weak Stationarity

- $E(x_t) = \mu$, i.e. the mean does not change over time
- $Cov(x_t, x_{t+k}) = \gamma(k)$
    - Called the autocovariance function (ACVF), suggests that it is independent of t
    - $\gamma(0) = var(x_t)$, independent of t
    - The covariance between x_t and x_{t+k} does not change over time
    - $-\gamma(-k) = \gamma(k)$

(5.3.2) Strong Stationarity

The process $\{x_t : t = 0, \pm1, \pm2, \ldots\}$ is strictly (strong) stationary if the distribution of

$$(x_1, \ldots, x_j) \text{ and } (x_{t+1}, \ldots, x_{t+j})$$

are the same for any k and t

This means that strong stationarity implies weak stationarity.

(5.3.3) Autocorrelation Function (ACF)

In order to confirm that stationarity is apparent in the model, exploratory data analysis is carried out on the datasets, firstly with visual inspection of the plot, and then the autocorrelation function (ACF) plot is examined, with the ACF defined as:

$$\begin{aligned}\rho(k) &= Corr(x_{t+k}, x_t) \\ &= \frac{Cov(x_{t+k}, x_t)}{\sqrt{Var(x_{t+k})Var(x_t)}} \\ &= \frac{\gamma(k)}{\gamma(0)} \quad \text{for stationary time series} \end{aligned} \tag{5.1}$$

The ACF:

- Measures linear relations of x_t and x_{t+k};
- $-1 \leq \rho(k) \leq 1$, and also $\rho(k) = \rho(-k)$;
- is a semi-positive definitive function

If the plot of the ACF is found to be decreasing to zero at a significant (preferably exponential) rate, then it can be determined that the series is stationary.

(6) Simulation Studies

Now after providing some insight into the methods that are going to be used, it is possible to complete a series of simulation studies to compare the performance of each of the methods. In each of the studies, "empirical associations" will be found from each of the methods, which can then be compared with the true structure of the graphical model from which the data is simulated. In terms of the GMTS and SIN methods, the use of the different significance levels α will be considered for the GMTS approach, and then compared with the SIN method. The ℓ_1-regularization methods all use a significance level of 0.05 .This also gives some insight into how the Structural VAR models can be identified in the context of graphical models.

As stated before, the t-values that are associated with the GMTS approach are based off attempting to control the Family Wise error rate (FWER). The significance levels chosen are provided in the next table:

α value	Adjustment method
$\alpha = 0.05$	Unadjusted, set at $\alpha = 0.05$
α/M	Bonferroni adjusted

Table 6.1: Significance level for Unadjusted and Bonferroni-adjusted GMTS. M is the number of comparisons

To reiterate, M stands for the number of comparisons being made. The t-value that is used will then change depending on the α value used. In the table of results, T_{ua} will be the critical value under the unadjusted α value, while T_{Bo} signifies the Bonferroni adjusted α value. The critical values for each sample size are also provided in the table.

The measures that will be used to compare the methods will now be provided

(6.1) Statistics for Comparison:

Statistics	Description	Calculation
TPR	True positive rate, the proportion of edges in the estimated model that are also edges in the true model	TPR = TP /(TP + FN)
TNR	True negative rate, the proportion of edges excluded in the estimated model, that are also exclude in the true model	TNR = TN /(TN + FP)
FPR	False positive rate, the proportion of edges in the estimated model that have been misidentified	FPR = FP / (FP + TN)
FNR	False negative rate, the proportion of omitted edges that are present in the true model.	FNR = FN / (FN + TP)

Table 6.2: Measuring statistics for model performance.

A confusion matrix can be constructed using this information, with the knowledge that a 1 corresponds to an "edge" and a 0 corresponds to a "missing edge"

True	Estimated	
	0	1
0	TN	FP
1	FN	TP

Table 6.3: Confusion matrix

Another statistic can be used to summarize these values into a single value. Taking more of a look into the measuring statistics, where of the positive and negative rates has a name defining it:

- The True positive rate (TPR) is also called Sensitivity, or Recall
- The True negative rate (TNR) can be defined as Specificity
- The False positive rate is also named the Fall-out
- Finally the False negative rate (FNR) is called the Miss rate.

This allows us to implement the F_1 score (also referred to as the F-score or F-measure). This is a measure of a test's accuracy. In this case, it considers both the precision (also called the positive predictive value), which is the fraction of positive outcomes from the test that are actually true, i.e.:

$$precision = \frac{\Sigma\, True\ positive}{\Sigma\, Test\ outcome\ positive} \tag{6.1}$$

The recall, as previously defined, is also involved in the statistics. This statistics can be interpreted as a weighted average between recall and precision. Like the positive and negative rates, the F_1-score's best value is 1, and its worst value is 0.

After this has been calculated, the F_1-score can be calculated by:

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \tag{6.2}$$

In the context of the results produced in these analyses, this can also be defined as:

$$F_1 = \frac{2TPR}{2TPR + FPR + FNR} \tag{6.3}$$

Therefore this score can easily be calculated from the results from the positive and negative rates.

Now it is possible to introduce the simulations that were carried out. The first involved simulating a structural VAR(2) process with 3 time series; and the second model was a structural VAR(3) with 7 time series. Using a pre-defined adjacency matrix which showed the conditional dependencies and independencies, it was possible to use each of the methods to estimate the structure from the subsequently simulated datasets. Datasets of varying sample sizes (n = 1000, 2000, 5000, 10000, 20000, 50000) were simulated from the true models, where the data was simulated from a multivariate normal distribution. The four statistics, which were stated before, could then be calculated

(6.2) Deriving the CIG with the different approaches

It is now possible to deriving the CIG for each of the datasets using the different methods. As a note for the results, the original results showing the covariance/precision matrices will all be originally specified as 9x9 for the SVAR(2), and 28x28 for the SVAR(3). After finding the estimated partial correlation matrices, the aim is then to reduce down these matrices to 9x3 and 28x7. This is because only the conditional dependencies between the contemporaneous variables, and the relationships between the

contemporaneous and lagged variables are considered. Associations between solely the lagged variables are not considered in this analysis. After stating this, it can now be shown how each of the methods provides the results for the CIGs

(6.2.1) GMTS method

When using the GMTS method to produce the CIG, cigts, a MATLAB program written by M. Reale, is used. This program requires the full dataset, the number of lags, and the significance level, which is 0.05 for the unadjusted approach, and 1-(0.05/M) for the Bonferroni-adjusted approach. The t-values produced correspond to the critical value calculated as a result of the critical value, for example the SVAR(2) model with 1000 observations has a critical value of $T_{ua} = 1.96$ for the unadjusted approach, and $T_{Bo} = 3.216$. The output Rsig provides an adjacency matrix, where the significant partial correlations are indicated with a 1, and the insignificant partial correlations are given a 0.

A short note on the ℓ_1-regularization and SIN methods is required now, in terms of transforming the data. The GMTS approach has the added benefit of transforming the data so that the lagged variables are also included in the dataset. This is required so that the partial correlation matrices can become 9x9 for the SVAR(2), or 28x28 for the SVAR(3), which will then be transformed into the 9x3 and 28x7 matrices which are shown later in the section as displayed by table 6.4, with the graph of the SVAR(3) shown in figure 6.4

(6.2.2) SIN method

The R package SIN is used to derive CIGs for this analysis. The function sinUG computes a matrix of simultaneous p-values for the SIN model selection, requiring the sample correlation matrix, and the sample size. The function getgraph was then implemented, which compares the p-values computed from sinUG, from a pre-specified significance level, set at 0.025 in this case because this is a two-tailed test. A 9x9 adjacency matrix is then provided, where a 9x3 matrix showing the relationships between the contemporaneous variables and the contemporaneous and lagged variables is then extracted.

(6.2.3) glasso method

It is possible to use the glasso method in either R or MATLAB, but for this analysis, the results were produced in MATLAB. The program requires:

- The covariance matrix of the dataset,
- The tuning parameter λ,
- The maximum number of iterations the algorithm is allowed to carry out the estimations of the precision matrix (set at 5000 for the SVAR(2), and 15000 for the SVAR(3)
- The convergence tolerance level, which is set at $1*10^{-16}$ (machine accuracy).

The program then outputs the final estimates of the covariance and precision matrices.

(6.2.4) SPACE method

The R package SPACE is used to produce the adjacency matrices in this case. The function used in this case is "space.joint", which estimates the partial correlations using the joint sparse regression model. The inputs required are:

- The transformed dataset
- The ℓ_1 penalty used in the method
- The number of iterations, which will be provided for each dataset in the table 6.4

The number of iterations used in this approach is important, because it was shown in the analysis that it is not always the best idea to have the process to have too many iterations. This is because it was found that the true positive and negative rates diminished slightly if too many iterations were used. An optimal number of iterations were found for each dataset. The important output from this function is then the partial correlation matrix, which can then be transformed into an adjacency matrix by comparing the individual partial correlations with a significance level specified previously ($\alpha = 0.05$).

(6.2.5) CLIME method

Both the CLIME method and the TIGER method have their functions contained in the R package flare. For the CLIME method, there are two steps to computing the precision matrices which are subsequently used to produce the adjacency matrices. The first step uses the function sugm, which requires:

- The transformed data matrix
- lambda, which is set to default in this case. This then allows R to compute the tuning parameters based off nlambda and lambda.min.ratio

- nlambda, which is the number of lambda values that are used in the analysis, this is set to 20.
- lambda.min.ratio is the minimum lambda value considered in the analysis. This is set to machine accuracy ($1*10^{-16}$).

The second function used is sugm.select, which helps to provide the optimal covariance matrix based off the lambda values provided. This requires:

- The output from the sugm function
- The criterion used to determine the optimal matrix. Cross-validation is used in this analysis.
- The number of folds used in the cross-validation.

After these two steps, it is then possible to produce the optimal precision matrix estimate according to the method, which can then be used to produce the adjacency matrices.

(6.2.6) TIGER method

Like the CLIME method, the TIGER approach uses the function sugm. Only this time, the output requires

- The transformed dataset
- the tuning parameter, which was set at $\sqrt{(9)}/n$

This then produced the precision matrix estimates which were used to produce the adjacency matrix for each dataset.

(6.3) SVAR(2) model

Referring back to the section describing the structural VAR model, it is easy to understand that the SVAR(2) can be modelled as:

$$A_0X_t = A_1X_{t-1} + A_2X_{t-2} + \varepsilon_t \qquad (6.4)$$

where the coefficient matrices are

$$A_0 = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \qquad A_1 = \begin{pmatrix} 0.9 & 0 & 0 \\ 0 & 0.6 & 0 \\ 0 & 0 & -0.5 \end{pmatrix} \qquad A_2 = \begin{pmatrix} -0.6 & 0 & 0 \\ 0.9 & 0 & 0.4 \\ 0 & 0 & 0 \end{pmatrix}$$

and the covariance matrix of the error terms ε_t are:

$$\Sigma_\varepsilon = \begin{pmatrix} 0.09 & 0 & 0 \\ 0 & 0.04 & 0 \\ 0 & 0 & 0.025 \end{pmatrix}$$

The model that is considered in the simulation studies is defined as:

$$\begin{pmatrix} x_t \\ y_t \\ z_t \end{pmatrix} \begin{pmatrix} 0.9 & 0 & 0 \\ 0.9 & 0.6 & 0 \\ -0.9 & 0 & -0.5 \end{pmatrix} \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ z_{t-1} \end{pmatrix} \begin{pmatrix} -0.6 & 0 & 0 \\ 0.3 & 0 & 0.4 \\ 0.6 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_{t-2} \\ y_{t-2} \\ z_{t-2} \end{pmatrix}$$

Which is the moralized version of equation 6.1. Here we denote the variables as x, y, z. There are three contemporaneous variables x_t, y_t, and z_t, and each has two lagged variables, x_{t-1} and x_{t-2}, y_{t-1} and y_{t-2}, and z_{t-1} and z_{t-2}. The stationarity has been studied for this model beforehand (See appendix for plot and output from the tests). The figure below shows the true CIG.

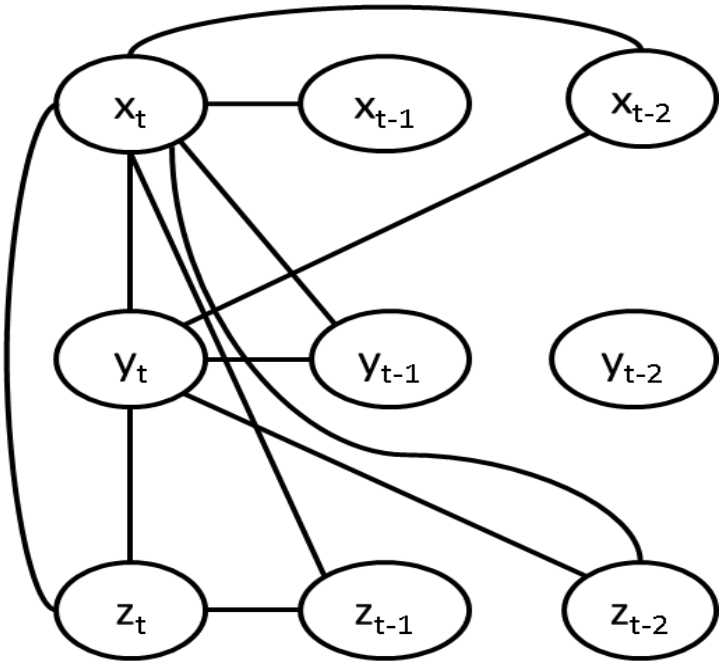


Figure 6.1: True CIG for SVAR(2) model

Variable	x_t	y_t	z_t
x_t	1	x	x
y_t	1	1	x
z_t	1	0	1
x_{t-1}	1	0	0
y_{t-1}	1	1	0
z_{t-1}	1	0	1
x_{t-2}	1	1	0
y_{t-2}	0	0	0
z_{t-2}	1	1	0

Table 6.4: true adjacency matrix for SVAR(2)

The adjacency matrix is on the right, beside the CIG. A 1 corresponds to an edge between two variables, and a 0 indicates no edge. The first column shows the 9 variables being examined, with columns 2 to 4 showing the conditional dependencies between the contemporaneous and lagged variables, and the contemporaneous variables themselves. The x's mean that this element does not need to be considered, due to the symmetric nature of the partial correlation matrix. For example, the edge between x_t and y_t is already shown in (2,1), so there is no need to "re-estimate" the correlation for element (1,2).

(6.3.1) Results

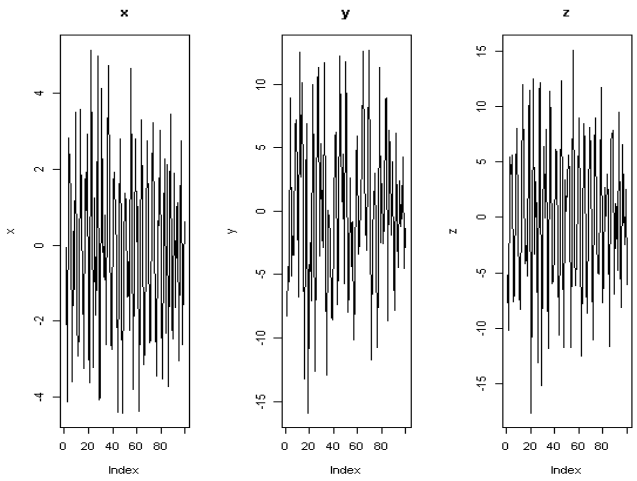


Figure 6.2: Plots of the three time series for the SVAR(2)

Sample size	Method	TPR	TNR	FPR	FNR
1000	T_{ua}=1.965	0.9564	0.9380	0.0620	0.0436
	T_{Bo}=3.216	0.9293	0.9950	0.0050	0.0707
	SIN	0.929	0.998	0.002	0.071
	glasso	0.9	0.8889	0.1111	0.1
	SPACE (4)	0.9943	0.983	0.017	0.0057
	CLIME	0.7836	0.299	0.701	0.2164
	TIGER	0.5557	0.9	0.1	0.4443
2000	T_{ua}=1.961	0.9764	0.9520	0.0480	0.0236
	T_{Bo}=3.202	0.9443	0.9980	0.0020	0.0557
	SIN	0.944	0.997	0.003	0.056
	glasso	0.9133	0.9111	0.0889	0.0867
	SPACE (4)	0.9886	0.993	0.007	0.0114
	CLIME	0.7864	0.286	0.714	0.2136
	TIGER	0.5714	0.9	0.1	0.4286
5000	T_{ua}=1.96	0.9986	0.9520	0.0480	0.0014
	T_{Bo}=3.2	0.9829	0.9990	0.0001	0.0171
	SIN	0.981	1	0	0.019
	glasso	0.9067	1	0.0	0.1
	SPACE (4)	0.9886	1	0	0.0114
	CLIME	0.8007	0.4	0.6	0.1993
	TIGER	0.6114	0.897	0.103	0.3886
10000	T_{ua}=1.96	1	0.9520	0.0480	0
	T_{Bo}=3.2	0.9993	0.9970	0.0030	0.0007
	SIN	0.999	0.997	0.003	0.001
	glasso	0.9	1	0	0.1
	SPACE (4)	0.9821	1	0	0.0179
	CLIME	0.7779	0.439	0.561	0.2221
	TIGER	0.5379	0.784	0.216	0.4621
20000	T_{ua}=1.96	1	0.9560	0.0440	0
	T_{Bo}=3.2	1	0.9980	0.0020	0
	SIN	1	0.998	0.002	0
	glasso	0.9	0.8889	0.1111	0.1
	SPACE (4)	0.985	1	0	0.015
	CLIME	0.7136	0.565	0.435	0.2864
	TIGER	0.5	0.714	0.286	0.5
50000	T_{ua}=1.96	1	0.9530	0.0470	0
	T_{Bo}=3.2	1	0.9990	0.0010	0
	SIN	1	0.999	0.001	0
	glasso	0.9267	1	0	0.0733
	SPACE (4)	0.9943	1	0	0.0057
	CLIME	0.6379	0.78	0.22	0.3621
	TIGER	0.5	0.7	0.3	0.5

Table 6.5: Model performance of SVAR(2)

Sample size	Method	F_1-score
1000	T_{ua}=1.965	0.9477
	T_{Bo}=3.216	0.9609
	SIN	0.9622
	glasso	0.8950
	SPACE (4)	0.9907
	CLIME	0.6308
	TIGER	0.6713
2000	T_{ua}=1.961	0.9646
	T_{Bo}=3.202	0.9704
	SIN	0.9564
	glasso	0.9123
	SPACE (4)	0.9928
	CLIME	0.6290
	TIGER	0.7133
5000	T_{ua}=1.96	0.9759
	T_{Bo}=3.2	0.9913
	SIN	0.9904
	glasso	0.9477
	SPACE (4)	0.9943
	CLIME	0.6671
	TIGER	0.7133
10000	T_{ua}=1.96	0.9766
	T_{Bo}=3.2	0.9982
	SIN	0.9980
	glasso	0.9474
	SPACE (4)	0.9901
	CLIME	0.6642
	TIGER	0.6134
20000	T_{ua}=1.96	0.9785
	T_{Bo}=3.2	0.9990
	SIN	0.9990
	glasso	0.8950
	SPACE (4)	0.9924
	CLIME	0.6642
	TIGER	0.5599
50000	T_{ua}=1.96	0.9770
	T_{Bo}=3.2	0.9995
	SIN	0.9995
	glasso	0.9620
	SPACE (4)	0.9971
	CLIME	0.6867
	TIGER	0.5556

Table 6.6: F_1-scores of SVAR(2)

Table 6.5 shows the results from the VAR(2) model. It is set out so that the different sample sizes are represented in the first column, and then each of the 7 methods implemented is displayed in the second column. The true positive and negative rates are shown in columns 3 and 4, while the false positive and negatives are finally shown in columns 5 and 6. The t_{ua} corresponds to the unadjusted significance value α (which is

set to 0.05 in these simulations) for the GMTS approach, and T_{Bo} corresponds to the Bonferroni adjusted GMTS approach. SIN refers to the SIN method proposed by Drton and Perlman. Finally the last 4 rows of each scenario, glasso, SPACE, CLIME, and TIGER, refer to the 4 ℓ_1-regularization methods that have been examined in this project.

Because the T_{Bo} requires a more significant level of conditional dependency to be regarded as an edge between 2 variables in the estimated model, there is expected to be a better true negative rate than for the T_{ua} approach, at the expense of the true positive rate. It can also be noted that using this method provides remarkably similar results to the SIN method also implemented.

An interesting aspect to examine is to see whether the ℓ_1 -regularization methods yield improved results over the previous methods implemented. Looking at the glasso and SPACE methods, there is a case to suggest that this aspect may be true, especially in terms of finding the conditional independencies between the variables, with SPACE in particular yielding perfect classification of the excluded as the sample size increases.

However, examining the CLIME and TIGER methods, very curious conclusions can be drawn from the results given. It is usually expected that as the sample size from a simulation increases, the datasets created should more closely reflect the true model from which the simulation is derived, and therefore the results should improve as well for the methods used. This belief is confirmed, more or less, in the first 5 methods considered (and perhaps there is the case for this in the CLIME method), but for the TIGER method, the results seem to deteriorate with increased sample size. A small study will be made into the TIGER method later in the discussion section. In terms of the CLIME method, the average true negative rate is very poor with a small sample size, but improves as the sample sizes increase. However, this is at the expense of the true positive rate, which decreases with increased sample size.

Table 6.6 provides the F_1-scores from each of the methods. This helps to give a clearer understanding of which method performs the best according to the simulated dataset, compared to the true positive and negative rate measures. The top six methods (T_{ua} to CLIME in the tables), all increase in F_1-scores at varying rates, with increasing sample sizes. This provides a clearer conclusion for the CLIME method, where it was not obvious whether this had occurred. The TIGER method has been shown, using this measure, to deteriorate with increasing sample size.

(6.4) SVAR(3) model.

The structural VAR(3) can be represented as:

$$A_0X_t = A_1X_{t-1} + A_2X_{t-2} + A_3X_{t-3} + \varepsilon_t \quad (6.5)$$

With the coefficient matrices A_o, …, A_3:

$$A_0 = \begin{pmatrix} 1 & & & & & & \\ 0 & 1 & & & & & \\ 0 & 0 & 1 & & & & \\ -2.73 & 0.38 & -3.18 & 1 & & & \\ -0.36 & 0 & 0 & 0.05 & 1 & & \\ 1.56 & -0.47 & 0 & -0.5 & -1.93 & 1 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_1 = \begin{pmatrix} 0.9048 & 0.6 & 0 & 0 & 0.28 & 0 & 0 \\ -.1898 & -0.52 & -0.3998 & 0.07 & 0 & -0.02 & 0 \\ 0.02 & -0.12 & -0.4 & 0 & -0.08 & 0 & 0 \\ 0.6398 & 0 & 0 & 0 & 0 & -0.21 & 0 \\ 0.3 & 0.02 & 0 & 0 & -0.91 & 0 & 0.21 \\ -2.0098 & 0 & -0.5402 & 0 & 1.9391 & 0.13 & 0 \\ 0.1 & 0 & 0 & 0.3 & 0 & 0 & -0.31 \end{pmatrix}$$

$$A_2 = \begin{pmatrix} -0.2 & 0.012 & 0 & 0 & 0 & 0 & -0.23 \\ -0.0351 & -0.0344 & -0.0834 & 0 & -0.2316 & 0.12 & -0.05 \\ 2.6108 & -0.3983 & 1.9175 & -0.783 & -0.6948 & 0.36 & -0.047 \\ 0.3549 & 0 & 0.0834 & -0.13 & 0 & 0 & 0 \\ -0.15 & 0.1688 & -0.39 & 0 & 0 & 0 & 0 \\ 1.2555 & -0.329 & 1.749 & -0.55 & 0 & 0 & 0 \\ -0.1998 & 0 & 0 & 0 & -0.05 & -0.1 & 0 \end{pmatrix}$$

$$A_3 = \begin{pmatrix} -1.0238 & 0.114 & -0.954 & 0.3 & -0.28 & 0 & 0.63 \\ 0.334 & 0.52 & 0.4 & -0.09 & -0.4 & 0 & 0 \\ -0.0576 & 0.12 & 0.6 & 0 & -0.08 & 0 & 0 \\ -0.5679 & 0 & 0.2001 & -0.0101 & -0.2003 & 0.21 & 0 \\ -0.1128 & 0 & -0.1 & -0.026 & 0.39 & 0 & -0.21 \\ 0.1874 & 0 & 0.5404 & 0 & -0.7 & -0.13 & 0 \\ 0.48 & 0 & 0.04 & -0.325 & -0.5 & 0 & 0.31 \end{pmatrix}$$

The covariance matrix of ε_t merely has 0.19, 0.54, 0.15, 0.63, 0.13, 0.33, and 0.57 along the main diagonal, and 0 everywhere else. The variables used in this SVAR model are s,

u, v, w, x, y, and z. The CIG representation of the model can be seen below, as well as the individual time series in the model.

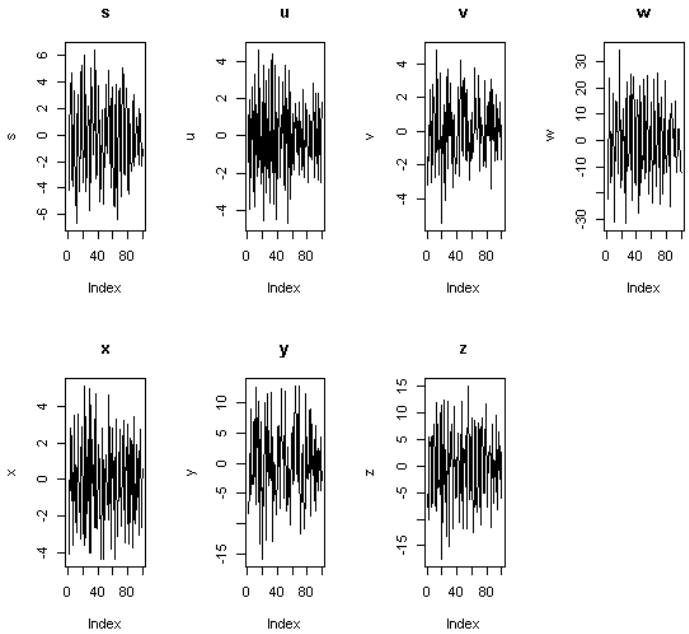


Figure 6.3: Plots of time series for SVAR(3)

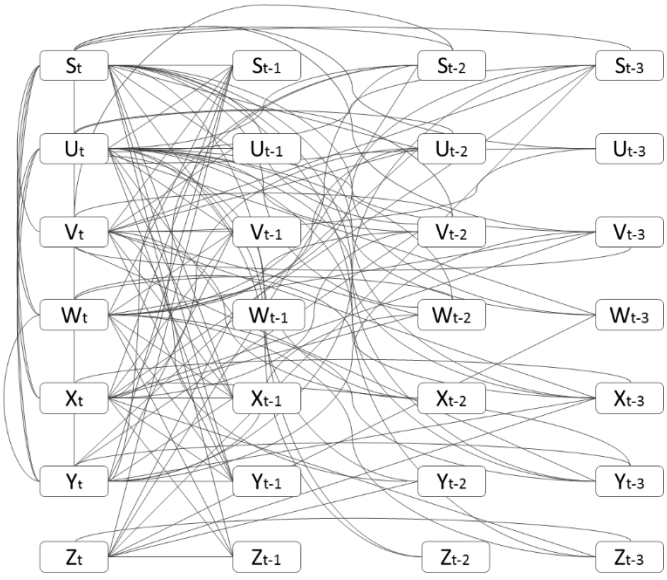


Figure 6.4: True CIG of SVAR(3)

(6.4.1) Results

n	Method	Ave[TPR]	Ave[TNR]	Ave[FPR]	Ave[FNR]
1000	T_{ua}=1.965	0.9648	0.7705	0.2295	0.0352
	T_{Bo}=3.216	0.9966	0.6285	0.3715	0.0034
	SIN	0.9963	0.6385	0.3615	0.0037
	Glasso	0.8534	0.7377	0.2623	0.1466
	SPACE	N/A	N/A	N/A	N/A
	CLIME	N/A	N/A	N/A	N/A
	TIGER	0.8086	0.736	0.264	0.1914
2000	T_{ua}=1.961	0.9642	0.8358	0.1642	0.0358
	T_{Bo}=3.202	0.9912	0.7140	0.2860	0.0088
	SIN	0.9910	0.7235	0.2765	0.0090
	Glasso	0.8897	0.9396	0.0604	0.1103
	SPACE	N/A	N/A	N/A	N/A
	CLIME	N/A	N/A	N/A	N/A
	TIGER	0.7079	0.769	0.231	0.2921
5000	T_{ua}=1.96	0.9588	0.9116	0.0884	0.0412
	T_{Bo}=3.2	0.9883	0.8051	0.1949	0.0117
	SIN	0.9877	0.8197	0.1803	0.0123
	Glasso	0.8983	0.9679	0.0321	0.1017
	SPACE	N/A	N/A	N/A	N/A
	CLIME	N/A	N/A	N/A	N/A
	TIGER	0.7757	0.699	0.301	0.2243
10000	T_{ua}=1.96	0.9543	0.9530	0.0470	0.0457
	T_{Bo}=3.2	0.9824	0.8707	0.1293	0.0176
	SIN	0.9816	0.8849	0.1151	0.0184
	Glasso	0.8957	0.9755	0.0245	0.1043
	SPACE	N/A	N/A	N/A	N/A
	CLIME	N/A	N/A	N/A	N/A
	TIGER	0.7786	0.685	0.315	0.2214
20000	T_{ua}=1.96	0.9508	0.9702	0.0298	0.0492
	T_{Bo}=3.2	0.9766	0.9283	0.0717	0.0234
	SIN	0.9763	0.9365	0.0635	0.0237
	Glasso	0.8371	0.7472	0.2528	0.1629
	SPACE	N/A	N/A	N/A	N/A
	CLIME	N/A	N/A	N/A	N/A
	TIGER	0.75	0.625	0.375	0.25
50000	T_{ua}=1.96	0.9440	0.9843	0.0157	0.0560
	T_{Bo}=3.2	0.9732	0.9625	0.0375	0.0268
	SIN	0.9727	0.9702	0.0298	0.0273
	Glasso	0.9017	0.9774	0.0226	0.0983
	SPACE	N/A	N/A	N/A	N/A
	CLIME	N/A	N/A	N/A	N/A
	TIGER	0.6421	0.7	0.3	0.3579

Table 6.7: Model performance of SVAR(3)

Sample size	Method	F_1-score
1000	T_{ua}=1.965	0.8794
	T_{Bo}=3.216	0.8417
	SIN	0.8451
	glasso	0.8067
	SPACE	N/A
	CLIME	N/A
	TIGER	0.7803
2000	T_{ua}=1.961	0.9060
	T_{Bo}=3.202	0.8705
	SIN	0.8741
	glasso	0.9125
	SPACE	N/A
	CLIME	N/A
	TIGER	0.7566
5000	T_{ua}=1.96	0.9367
	T_{Bo}=3.2	0.9054
	SIN	0.9112
	glasso	0.9304
	SPACE	N/A
	CLIME	N/A
	TIGER	0.7471
10000	T_{ua}=1.96	0.9537
	T_{Bo}=3.2	0.9304
	SIN	0.9363
	glasso	0.9317
	SPACE	N/A
	CLIME	N/A
	TIGER	0.7438
20000	T_{ua}=1.96	0.9601
	T_{Bo}=3.2	0.9536
	SIN	0.9573
	glasso	0.8011
	SPACE	N/A
	CLIME	N/A
	TIGER	0.7059
50000	T_{ua}=1.96	0.9634
	T_{Bo}=3.2	0.9680
	SIN	0.9715
	glasso	0.9372
	SPACE	N/A
	CLIME	N/A
	TIGER	0.6612

Table 6.8: Model performance of SVAR(3) using F_1-score

The first issues to note here are with the SPACE and CLIME methods. Firstly for the SPACE method, finding the positive and negative rates required approximately 2 days for the n=1000 sample size alone. After this time, it was found that while the true positive rate was at 0.8890, the true negative was very low at 0.3081, after 6 iterations

of the process. This suggested that it was perhaps not useful to produce the results for the rest of the simulated datasets. This was because:

1. It suggests a very poor computational efficiency, which is an important concept that will be considered later in the discussion. As well as this, since the largest dataset was 50 times larger, it suggested a long time period of waiting to produce results.
2. While it may have been possible that the results from the other bigger datasets would have been significantly better, there are other methods available that produced more promising results for analysis.

In terms of the CLIME method, when running the program on the VAR(3) datasets in R, warning messages appeared stating that the estimated covariance matrices were singular, and a zero matrix was produced when inspections of the issues were made. The reason behind this is unknown, and requires further analysis.

In terms of the methods that we can check, it appears again that the previous methods implemented (those not requiring ℓ_1-regularization) provide the best results. It becomes more apparent here the difference between the Bonferroni-adjusted α and the unadjusted level, where surprisingly the true negative rates are inferior for the Bonferroni in this study. Again it can be seen that the results from SIN closely match those from the Bonferroni adjusted GMTS approach.

Again, the glasso provides a case for more analysis, providing either similar or comparable true negative rates in the majority of the simulations. In this case, there is some interesting behavior occurring for n=1000 and n=20000, which requires more analysis as well.

Finally, similar to the case of the VAR(2) model, the TIGER method provides inferior results compared to the rest of the methods. For unknown reasons, when the n=50000 dataset is used for analysis, the true positive rate drops significantly compared to the rest of the datasets. There is not a significant improvement on the true negative rate to account for this.

Table 6.8, which shows the F_1-scores, provides useful conclusions for the analysis. The first four methods are shown to improve with increasing sample size, except the dataset with 20000 observations, where the F_1-score drops by 0.13, which is a significant decrease in the context of the results. The TIGER, as in the SVAR(2) model, is shown to deteriorate with increasing sample size.

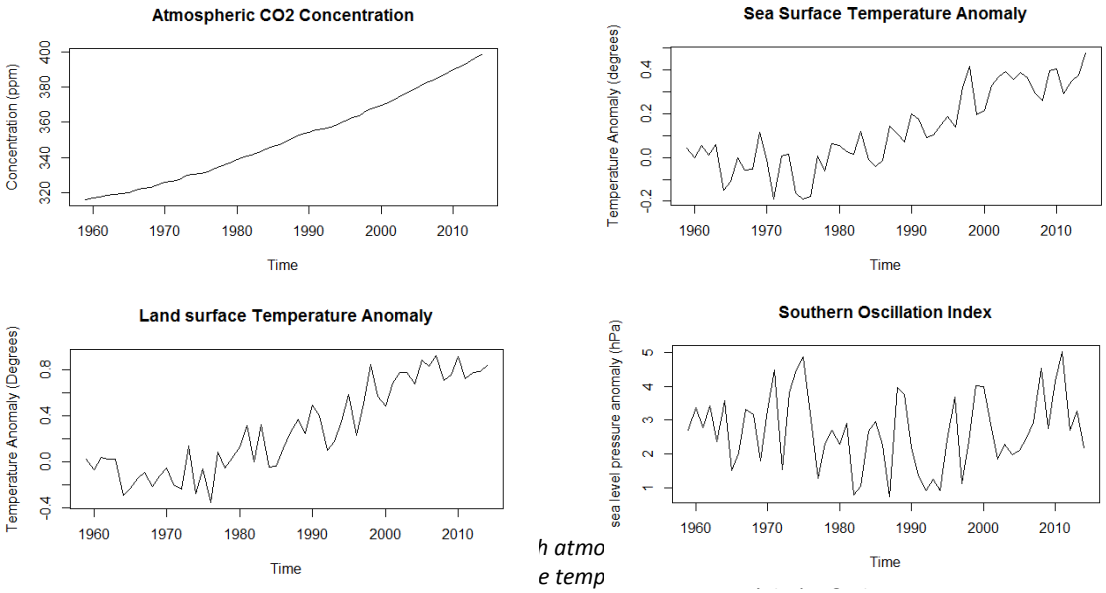
(6.5) Climate Data

These methods will now be implemented on a real dataset to see how the results compare. The dataset was provided courtesy of Wilson (2010), who uses the dataset in his own research in the implementation of structural vector autoregressive models in graphical modelling. It should be noted that the version of the paper used in this research is different from the paper available online. The version online provides different results to the results found in the version examined in this thesis.The data consists of four time series, with annual values provided for each time series from 1959 to 2014. These time series are:

1. The atmospheric carbon dioxide (CO_2) concentration in parts per million (ppm), observed at Mauna Loa, with data available from ftp://aftp.cmdl.noaa.gov/products/trends/co2/,
2. The global land surface temperature anomaly,
3. The global sea surface temperature anomaly,
4. The Southern Oscillation Index (SOI), which is the observed sea level pressure difference between Tahiti and Darwin, Australia, measured in hectopascals. The data is available from http://www.cpc.ncep.noaa.gov/data/indices/.

The global land and sea surface temperature anomalies are found from the Met Office Hadley Centre. The data for the global land surface and sea surface temperatures is available from http://www.cru.uea.ac.uk/cru/data/temperature/.

The figures below show the plots of each of the four time series.



h atmo
e temp
oscillation index (bottom right)

There are some expected dependencies occurring between these variables within one year. For example the CO_2 levels will have an influence on the air temperature, and also, the concentration of CO_2 from the oceans depends on the temperature of the sea surface. The structural vector autoregressive models used in the simulation studies were used to model these time series as a result of this, due to the inherent contemporaneous dependencies between the variables.

As a result, it is evident to remove the trends in each of the series, apart from the SOI time series, which does not exhibit one. Following the work of Wilson, the CO_2 series is corrected for a quadratic trend, while the two temperature series are corrected for a linear trend. Finally, the SOI time series is mean corrected, because trend correction does not make a significant difference to the time series. These plots are shown in the second section of plots.

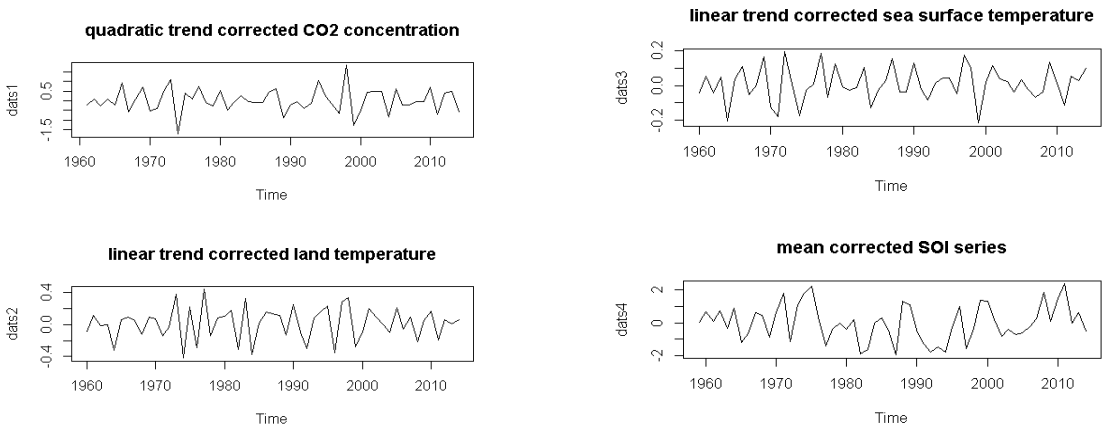


Figure 6.6: Trend and mean corrected time series for Climate data

The final section of plots show the autocorrelation function (ACF) plots for the four newly adjusted time series. In order to show stationarity in this scenario, the autocorrelation values must decay at a significant rate, often exponential is preferred, to zero. It should be noted that the first line of the ACF plot should always be one, because it occurs at k=0, therefore referring back to the equation for the autocorrelation function introduced in the stationarity section:

$$\begin{aligned}\rho(0) &= Corr(x_t, x_t) \\ &= \frac{\gamma(0)}{\gamma(0)} \\ &= 1\end{aligned}$$

Examining the ACF plots, it does appear that the time series are all stationary, since the autocorrelations do appear to decay at a significant rate below the significance line (the horizontal dotted blue line).

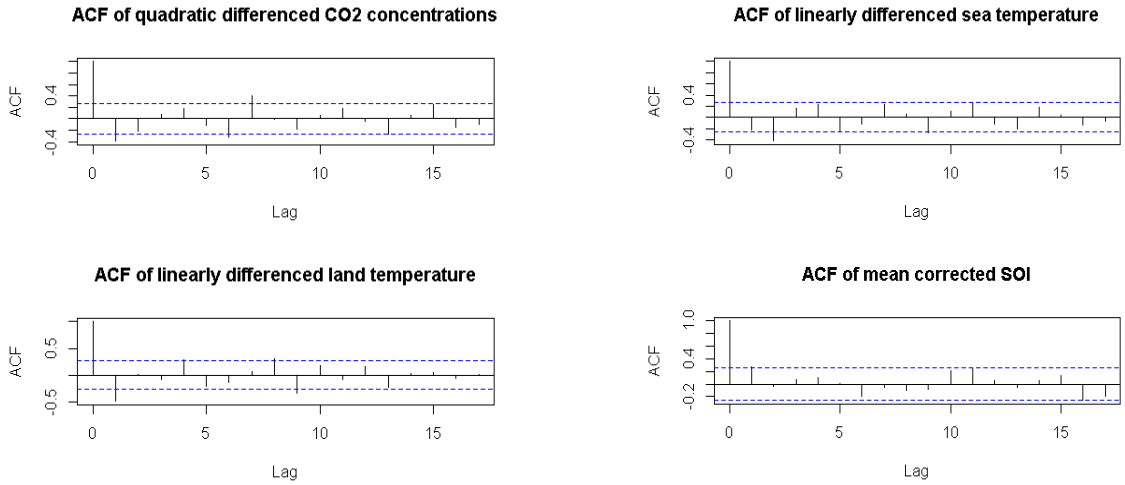


Figure 6.7: ACFs of corrected time series

Since structural VAR models are being implemented in this approach, it is important to determine the order p of the model. An Akaike Information Criterion (AIC) and its modified form (modified AIC) to determine the order of the VAR model is then used. An order of 2 was found to be the best choice for modelling the time series. The structural form is then fitted, in order to allow for contemporaneous conditional dependencies, which was alluded to previously.

Having achieved all of this, a test statistic is used to determine the significant partial correlations between variables, in a similar manner to the GMTS approach. A conditional independence graph (CIG) can then be drawn from these results, and is shown below in figure 6.8

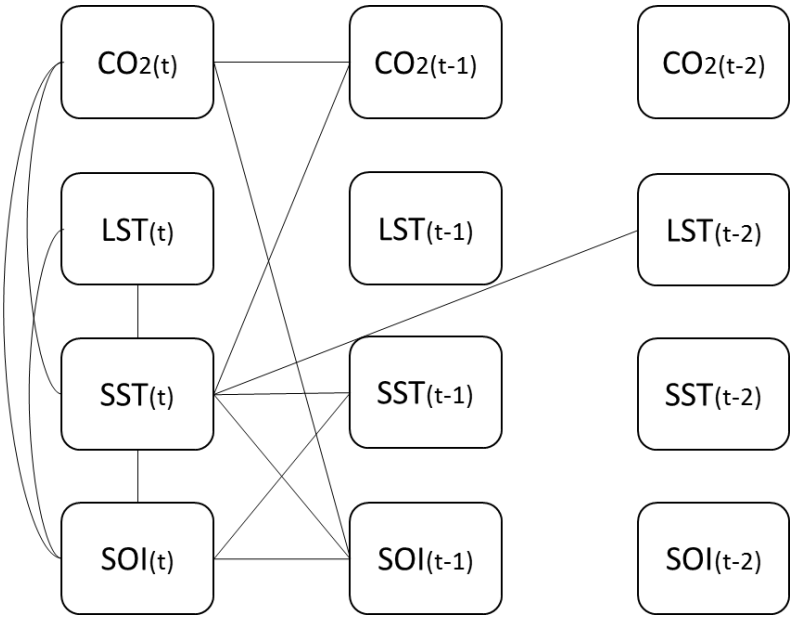


Figure 6.8: Estimated CIG of the Climate MTS, using the approach from the paper

The methods shown in this thesis can now be used on the same dataset (using the same preprocessing) to compare the graphs produce from the methods, with the graph provided by Wilson. A few interesting methods to include are the graphs produced from the GMTS method, including both the unadjusted and Bonferroni-adjusted graphs, and the glasso.

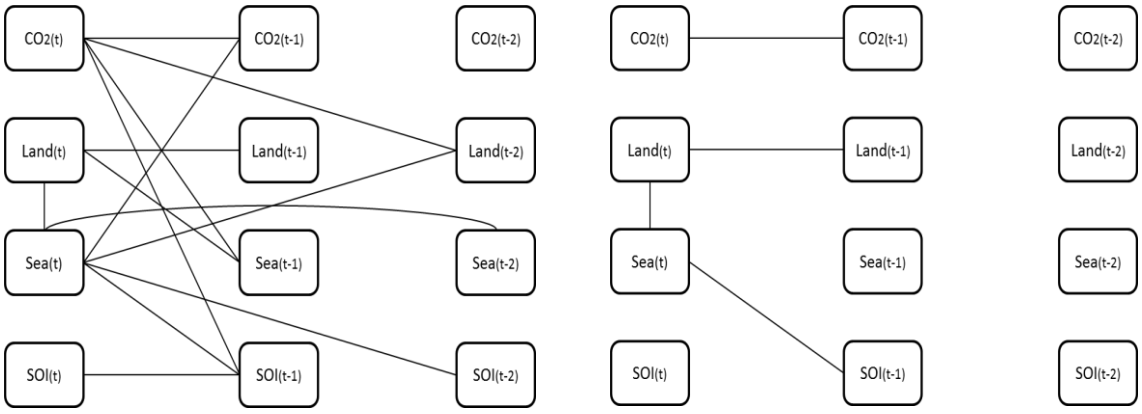


Figure 6.9: CIG of Climate MTS from GMTS unadjusted (left) and GMTS Bonferroni-adjusted (right)

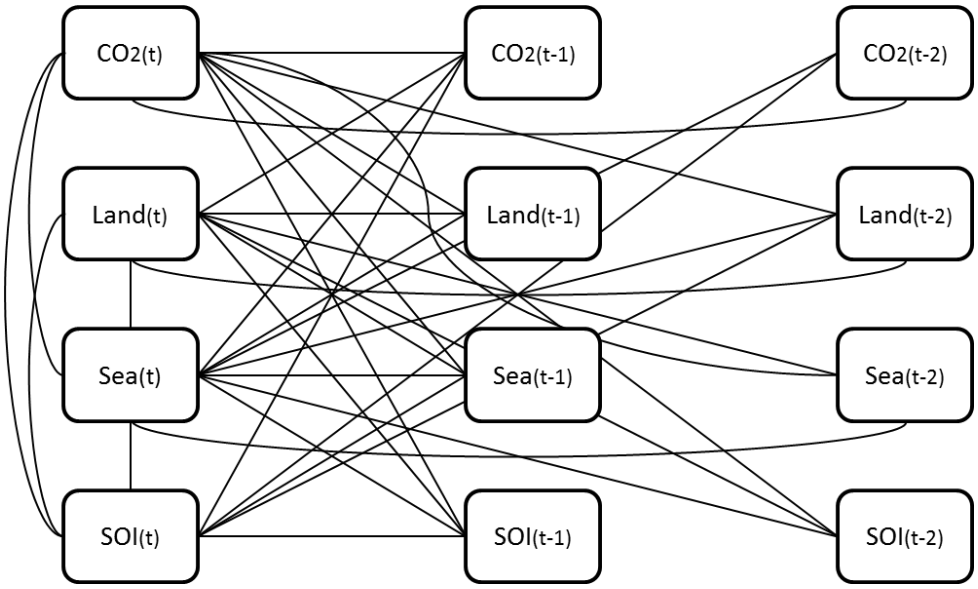


Figure 6.10: CIG of Climate MTS estimated using the glasso approach

Firstly, a comparison between the unadjusted GMTS approach and the Bonferroni-adjusted GMTS approach provides some significant differences. Prior analyses and knowledge on the Bonferroni adjustments gives insight into the reason that there are only three conditional dependencies occurring between variables, since the Bonferroni adjustment requires a much more significant conditional dependence between two variables to produce an edge between two variables. Curiously, now examining the differences in the graphs of the Wilson graph and the unadjusted GMTS graph, while there are more edges from the current time point to lagged variables, there are fewer links between contemporaneous variables in the GMTS approach, compared to the original graph. The glasso method in fact was the only method to provide the same edges between the contemporaneous variables as the original graph, although the glasso approach provides many more links in terms of connections to lagged variables.

The analysis of the residuals would give useful information about the adequacy of the different models. This however would involve the construction of a DAG and hence was not pursued in this thesis.

(7) Discussion

It is necessary to provide some discussions on not only the results found in this analysis, but also in terms of analysing the individual methods, with the other methods examined, and other methods available in other literature. Firstly, some issues associated with the glasso method will be discussed in section 7.1, with comparisons made with other modifications. Section 7.2 looks at the estimation and choices of the tuning parameters for each of the methods. Section 7.3 compares convergence rates between the methods considered, and also examines the convergence rates from other methods. Finally, section 7.4 provides a more overall comparison between methods, in terms of the motivations for using these methods.

(7.1) Glasso issues

The first paper examined is that by Mazumder and Hastie in 2012, which outlines some deficiencies associated with the glasso method. These issues will be discussed, and improvements are provided which, while they will not be used in this thesis, can be applicable to future research in the area.

The primary point to consider is to check what the glasso method is actually solving. The glasso approach and algorithm has been described in detail in previous sections, and it can be noted that in reality it solves the dual of the graphical lasso penalized likelihood, via block coordinate descent. For future reference, it is important to state that the objective of this method is to estimate the covariance matrix, Σ. Meanwhile, because the problem is partitioned (part of the block coordinate descent approach), while the whole covariance matrix changes from iteration to iteration, between iterations, only certain partitions are actually updated. The authors state that this accounts for a non-monotone behavior of the glasso method in minimizing the ℓ_1-penalized function. A corrected glasso block coordinate descent method, called "p-glasso" is created to amend for these issues. There is a warning with this method, the first being that there are rank-one updates required at each iteration, requiring another $O(p^2)$ operations, decreasing computational efficiency.

The next issue regards the estimation of the precision matrix Ω. As stated before, the inverse of the covariance matrix is not computed explicitly via the glasso method. Instead, it only keeps track of one partition of the matrix after every row/column update. This means this copy is not the exact inverse of the optimized variable Σ. Because this

method implements a block coordinate descent approach on the target matrix, the positive definiteness is retained in this matrix after every row/column update. However, because the estimated precision matrix Ω is not the exact inverse of the covariance matrix, it does not have to be positive definite. Rank one updates can be used after each iteration in order to provide an exact inverse of Σ, another issue arises from this, since this inverse does not have to be sparse. Amendments can be made to this to force some of the entries of the precision matrix to zero; but again, this may ruin the positive definiteness of the matrix. This is not an attractive prospect in ℓ_1-regularization methods.

Another concept that was considered by the authors was that of convergence, especially from a warm start. A warm start is the estimate of the covariance and precision matrix under the current penalization parameter λ_i, and using it in order to find the solution for λ_{i+1}. In an example provided, the estimated covariance matrix found from the glasso algorithm was found to contain a negative eigenvalue, which violates the positive definiteness of the matrix, and therefore affects convergence of the algorithm.

Another method, named "dp-glasso" accounts for the issues found throughout this thesis, providing a method that focusses on the estimation of Ω, and also promises sparsity and positive definiteness in the estimates. Also importantly, despite the added operations required to satisfy these conditions, the dp-glasso method is found to be faster than the glasso method.

(7.1.1) P-Glasso algorithm

First note that in this case, the precision matrix Ω is partitioned into blocks, as:

$$\Omega = \begin{pmatrix} \Omega_{11} & \Omega_{12} \\ \Omega_{21} & \Omega_{22} \end{pmatrix} \tag{7.1}$$

1. Initialized $W = \text{diag}(S) + \lambda I$, and the precision matrix estimate $\Omega = W^{-1}$
2. Cycle around the columns repeatedly, performing the following steps until convergence to the solution occurs:
    a. Rearrange the rows/columns so that the target column is last (implicitly).
    b. Compute $\Omega_{11}^{-1} = W_{11} - \frac{w_{12}w_{21}}{w_{22}}$
    c. solve

$$\min_{\alpha \in \mathbb{R}^{p-1}} \left\{ \frac{1}{2} \alpha' \Omega_{11}^{-1} \alpha + \alpha' s_{12} + \lambda \|\alpha\|_1 \right\} \tag{7.2}$$

for α, where $\alpha = \Omega_{12}w_{22}$. This is done using the solution from the previous round of updates is used as a warm start.

$$\hat{\Omega}_{12} = \hat{\alpha} w_{22} \tag{7.3}$$

$$\hat{\Omega}_{22} = \frac{1}{w_{22}} + \hat{\Omega}_{12}\Omega_{11}^{-1}\hat{\Omega}_{12} \tag{7.4}$$

are both then updated

d. Update Ω and W as in the glasso method, ensuring that $\Omega W = I_p$.

3. Output the solution Ω and its exact inverse W

(7.1.2) DP-Glasso algorithm

1. Initialize $\Omega = \text{diag}(S + \lambda I)^{-1}$
2. Cycle around the columns repeatedly, performing these steps until convergence occurs:
   a. Rearrange the rows/columns so that the target column is last
   b. Solve

$$\min_{\gamma \in \mathbb{R}^{p-1}} \frac{1}{2}(s_{12} + \gamma)'\Omega_{11}(s_{12} + \gamma); \tag{7.5}$$

   subject to $\|\gamma\|_\infty \leqslant \lambda$ for $\tilde{\gamma}$, and update

$$\hat{\Omega}_{12} = -\Omega_{11}(s_{12} + \tilde{\gamma})/w_{22} \tag{7.6}$$

   c. Solve for Ω_{22} using the equation

$$\hat{\Omega}_{22} = \frac{1-(s_{12}+\tilde{\gamma})'\hat{\Omega}_{12}}{w_{22}} \tag{7.7}$$

   d. Update the working covariance $w_{12} = s_{12} + \tilde{\gamma}$

Where $\tilde{\gamma} = \lambda\gamma$ is the estimate of the components wise signs of the precision matrix, i.e.

$$\gamma_{jk} = \text{sign}(\Omega_{jk}) \text{ if } \Omega_{jk} \neq 0$$
$$\gamma_{jk} \in [-1, 1] \text{ if } \Omega_{jk} = 0$$

(7.1.3) Symmetric glasso paper

The next paper, written by Friedman et al in 2010, provides a more rounded assessment of the graphical lasso methods in general, as well as some insight into the SPACE method. This paper examines both edge-sparse and node-sparse (graphical models that look at deleting all the edges associated with a given node) graphical models, but for the sake of comparisons with the graphical lasso, the edge-sparse approaches will be focused on. The other two approaches that are considered are the paired group lasso, and the symmetric lasso. The Meinhausen and Bühlmann (MB) approach was also used in the comparisons

The symmetric lasso is a method closely related to the SPACE method used in the simulation studies, which had the motivation of symmetrizing the MB-lasso approach. First, the conditional distribution of x_j given the rest of the variables $x_{\setminus j}$ must be defined as:

$$x_j|x_{-j} \sim N\left(\sum_{i\neq j} x_i\beta_{ij}, \sigma^{jj}\right) \tag{7.8}$$

where

$$\beta_{ij} = -\frac{\Omega_{ij}}{\Omega_{jj}} \qquad \text{and} \qquad \sigma^{jj} = \frac{1}{\Omega_{jj}} \tag{7.9}$$

In this method, it is assumed that the conditional distribution of each variable on the rest is linear, so the negative log-product-likelihood for these conditional distributions becomes

$$l(\Omega) = \frac{1}{2}\sum_{j=1}^{p}\left[Nlog\sigma^{jj} + \frac{1}{\sigma^{jj}}\left\|x_j - XB_j\right\|_2^2\right] \tag{7.10}$$

B_j here is a p-length vector with elements β_{ij}, except for a 0 in the jth position (given the term "pseudo log-likelihood). This can also be written as:

$$l\left(\tilde{\Omega}\right) = \frac{1}{2}\sum_{j=1}^{p}\left[Nlog\sigma^{jj} + \frac{1}{\sigma^{jj}}\left\|x_j - X\tilde{\Omega}_j B_j\right\|_2^2\right] \tag{7.11}$$

where $\tilde{\Omega}$ is symmetric with zero along the main diagonal. The way to estimate a sparse form of this is to solve

$$\min_{\tilde{\Omega},\{\sigma^{ii}\}_1^p} \frac{1}{N} l\left(\tilde{\Omega}\right) + \lambda\sum_{i<j}\left|\tilde{\Omega}_{ij}\right| \quad \text{subject to } \tilde{\Omega}_{ij} = \tilde{\Omega}_{ji} \tag{7.12}$$

The idea then becomes to implement a coordinate descent method for decreasing λ values on the log scale, and updating $\tilde{\Omega}_{ij}$ after each iteration.

The other method proposed in this paper was the paired group lasso, which is stated to be a more direct modification of the MB approach, based on the grouped lasso. Before analysis, the columns x_i of the data matrix have to be standardized to have zero mean and unit norm. The aim is then to solve

$$\min_B \frac{1}{2}\|X - XB\|_F^2 + \lambda\sum_{j<i}\left\|(\beta_{ij}, \beta_{ji})\right\| \tag{7.13}$$

where $\|.\|_F$ means a Frobenius norm is used.

The first way in which these methods were compared was to carry out a small simulation study to examine the times taken to produce results by each method. Based on varying sample sizes and number of parameters, timing results were provided for each approach. It was found that the SPACE method was the slowest out of the methods examined (which provides some insight into the poor timing performances for the VAR(3) simulations studies). A possible reason behind this, as provided by Friedman et al, was that it was the only method to carry out its processes using C and R, whereas the other methods were used in double precision Fortran. However, it is still stated that the computation efficiency of SPACE should be comparable with the symmetric lasso, but it is unknown what caused SPACE to be so slow in the studies. The glasso method was also very slow with respect to the introduced methods, in some cases around 20 times slower.

Performance measures were then used for various simulation studies to determine which model performed the best. The fractional area under the ROC curve, which starts from zero false positives up to nz false positives, relative to perfect classification (all true positives are correctly identified before any false positives. nz corresponds to the number of non-zero diagonals elements in the precision matrix, where z = p(p-1)/2 – nz zero elements. The measure is then given as:

$$AUC_f \frac{\int_0^{nz/z} t(f)df}{nz/z} \tag{7.14}$$

where f is the false positive rate from a given point on the ROC curve, and t(f) is the true positive rate at that point. Therefore, $AUC_f = 1$ gives perfect selection (t(f) = 1 for all f > 0), and for a random selection scenario of positives, (t(f) = f), $AUC_f = nz/z$

The approaches described above (excluding the SPACE method) were then compared with two other methods, the univariate correlation approach, and the statewise approach. The univariate correlation method simply ranked the off-diagonal elements of the sample correlation matrix on their absolute values in descending order. The non-zero elements of Ω were then identified in that order. The statewise method on the other hand was derived from the symmetric lasso log-likelihood criterion discovered before. In this case, each element is identified to be non-zero if the corresponding component of the gradient (derivative) of the log-likelihood is the largest in absolute value out of all the elements. The log-likelihood is subsequently minimized with respect to all of the non-zero elements, including the newest element.

Firstly, analyses of the speed of each method concluded that the univariate correlation was by far the fastest method, followed by the newly introduced paired group lasso and symmetric lasso. The MB and statewise approaches were found to be the next slowest, with glasso providing the worst results in terms of computation speed.

The most curious result comes next however, where results suggested that the univariate correlation and statewise methods dominated the other approaches. Three out of the four simulation studies concluded that the univariate correlation performed the best, while the statewise well and truly outperformed the other methods in the fourth study.

(7.2) Tuning Parameters

An issue that must be considered when it comes to comparing the methods implemented in this research is the tuning/penalty parameter used. While the ℓ_1-regularization approach is implemented in each of the approaches considered, the method in which each of the tuning parameters is estimated is different for approach. Therefore each of the methods will produce different results. Therefore in the following subsections, the choice of tuning parameter will be critiqued.

(7.2.1) CLIME parameter and the Glasso parameter

Cross validation is used to estimate the tuning parameter used in the methods, with the CLIME method using it as a constraint to produce a feasible set of maximum likelihood estimates, and the glasso method using it to induce a specific level of sparsity on the graphical model. The usefulness of this result comes via the realization that it is a data-driven approach, which uses a training set to determine the optimum penalty parameter, to use on the test set, or the data that will be used to determine the structure of the graphical model. This is a more attractive approach to those implemented in other methods, which use finite-sample or asymptotic theories in deriving the tuning parameters (Liu & Luo, 2012). More on this will be discussed later.

(7.2.2) SPACE parameter/glasso approach

The authors of the paper who introduced the SPACE approach decided upon using a Bayesian Information Criterion to deduce what the tuning parameter should be in the model. The reasoning for this was that it provides a simple and computationally easy way of determining the parameter. Because of the nature of the lasso method which this method borrows some methodology from, there is the opportunity to use different methods to estimate the tuning parameter. In the simulation studies, a mixture of cross-

validation and information criterion was used to determine the penalty parameter to use in the final method. In terms of the graphical lasso, whether it was due to the nature of the data, or the amount of penalization that occurred, the AIC suggested that the original maximum likelihood estimate (one that did not require a tuning parameter in it) was the best fit for the model. The graph below shows this, with a near inverse relationship between the log-likelihood and the AIC (since the parameters seemed to be included one-by-one). The Climate data application examined further emphasizes this point, even suggesting that a near full model, where all except one of the edges are statistically significant, provided the best fit to the model. When comparing with all of the other methods, a same conclusion is clearly not provided in any of the cases.

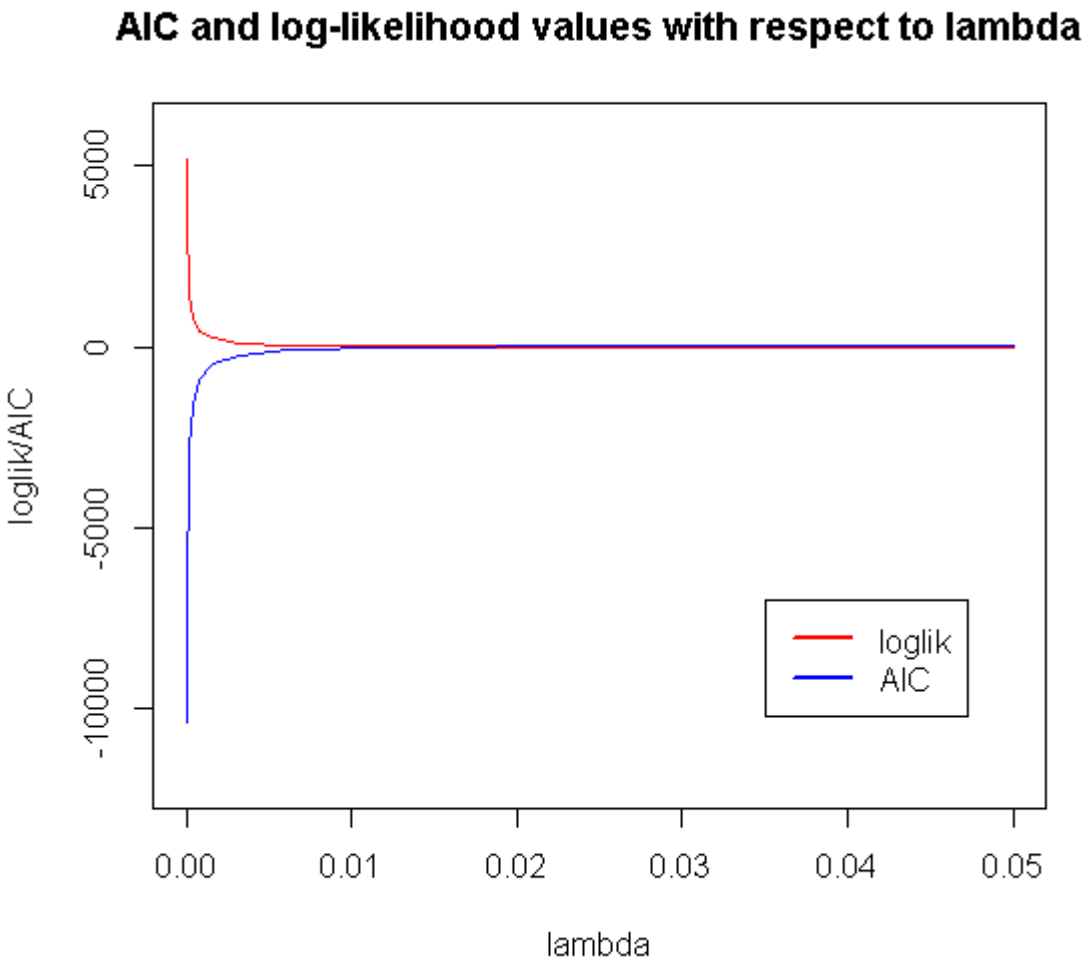


Figure 7.1: graph comparing AIC and log-likelihood values according to the choice of lambda. The red line corresponds to the log-likelihood values, while the blue line shows the AIC values.

(7.2.3) Dantzig vs. Lasso

Referring back to the flare paper that was discussed for the TIGER tuning parameter, it is revealed that the CLIME method uses the Dantzig selector, first introduced by

Candes and Tao in 2007, to provide the sparse structure to the graphical model. It may be interesting to have some insight into what makes the Dantzig selector different from the Lasso, which the graphical lasso in particular is modified from.

James et al, in 2009, proposed a new algorithm, called DASSO, which fitted the coefficient path of the Dantzig selector, but with a similar computational cost to the least angle regression algorithm that the lasso method uses. While writing this, the authors provide some connections between the two methods.

In terms of notation, the lasso estimate $\hat{\beta}_L = \hat{\beta}_L(\lambda_L)$ is defined as:

$$\hat{\beta}_L = arg \min_{\tilde{\beta}} (\frac{1}{2} \left\| Y - X\tilde{\beta} \right\|_2^2 + \lambda_L \left\| \tilde{\beta} \right\|_1) \tag{7.15}$$

This is equivalent to

$$\min (\left\| \tilde{\beta} \right\|_1) \quad \text{subject to } \left\| Y - X\tilde{\beta} \right\|_2^2 \leqslant s \tag{7.16}$$

for some non-negative s.

Next the Dantzig selector is defined to compare with the lasso. The solution is defined as:

$$\min (\left\| \tilde{\beta} \right\|_1) \quad \text{subject to } \left\| X^T (Y - X\tilde{\beta}) \right\|_\infty \leqslant \lambda_D \tag{7.17}$$

where λ_D is the tuning parameter for the Dantzig selector.

In light of these solutions, it appears there are similarities between the two approaches. In fact, the only difference is that the Dantzig selector penalizes $\left\| \tilde{\beta} \right\|_1$, the sum of absolute coefficient, by using the L_∞-norm of the p-vector $X^T(Y - X\tilde{\beta})$, while on the other hand, the lasso regularizes $\left\| \tilde{\beta} \right\|_1$ via the residual sum of squares

When the tuning parameters λ_L and λ_D are chosen to be the same, the lasso estimate is going to be always be a feasible solution to the Dantzig selector minimization problem, even though it may not be an optimal solution. If the optimal solutions are in fact not identical, then the Dantzig selector provides a sparser solution compared to the lasso estimate.

To further emphasize this similarity, it is possible to rewrite the solutions above as

$$\min (\left\| \tilde{\beta} \right\|_1) \quad \text{subject to } \left\| X^T X(\tilde{\beta} - \hat{\beta}_{ls}) \right\|_\infty \leqslant t_D \tag{7.18}$$

$$\min\left(\left\|\tilde{\beta}\right\|_1\right) \quad \text{subject to } \left\|X(\tilde{\beta}-\hat{\beta}_{ls}\right\|_2^2 \leqslant t_L \tag{7.19}$$

Where $\hat{\beta}_{ls}$ is the least squares estimate, and t_D and t_L are some constants.

Analyses have not been carried out to confirm that the Dantzig selector provides a more sparse solution under the same tuning parameter, but it provides an interesting discussion point in terms of how these regularization methods compare and contrast to each other. All that can be said regarding comparisons between the performances is that the glasso method appears to perform considerably better than the CLIME in the simulations used (and actually provides usable results in terms of the VAR(3) approach).

(7.3) Convergence rates

While the simulation studies have provided some insight into the abilities of each of the methods in question to provide the sparse structure of the graphical model, it isn't the only way to judge which method is best to implement. The speed or efficiency at which the solution is provided is another important factor, because it is not useful to have a method that, even though it may produce a very good result, can take days to produce a result, especially when other methods are available which produce the same results in a fraction of the time. In some of the papers examined, the method of finding the convergence rates is to consider $\widehat{\Omega}-\Omega$, which correspond to the error between the estimated precision matrix and the real precision matrix. For the sake of comparisons between approaches, the Spectral and Frobenius norms have been used. In terms of the Spectral norm, Liu and Luo (2012), state that it is a useful measure of the rate of convergence it implies the convergence rate of the eigenvalue and eigenvector, which is said to be necessary in principle component analysis. Similarly, Cai, Zhang, and Zhou (2010) state that the Frobenius norm is useful because it can be used to define the numerical rank of a matrix, which again, it turns out, is useful in principle component analysis.

(7.3.1) Comparing methods used:

Method	Frobenius norm	Spectral Norm
SPACE	N/A	$\sqrt{\frac{\log(n)}{n}}$
CLIME	$s\sqrt{\frac{\log(p)}{n}}$	$kM_p^2\sqrt{\frac{\log(p)}{n}}$
Glasso	$\sqrt{\left(\frac{p+s}{n}\right)\log(p)}$	$\sqrt{\left(\frac{s}{n}\right)\log(p)}$
TIGER	$\|\theta\|_1\sqrt{\left(\frac{p+s}{n}\right)\log(p)}$	$kM_p\sqrt{\frac{\log(p)}{n}}$

Table 7.1: Table of convergence rates for the methods considered under the Frobenius and Spectral norms

s = total number of non-zero off-diagonal elements of Ω ;

p = the number of parameters;

n = the number of observations;

$\|\theta\|_1$ = the matrix 1-norm of the precision matrix;

M_p = the upper bound of the 1-norm of the precision matrix i.e. $\|\theta\|_1 \leqslant M_p$;

and k is a constant.

To reiterate, Frobenius norm is defined by the square root of the sum of absolute squares of the elements of the matrix in question

$$\|A\|_F\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2} \tag{7.20}$$

The Spectral norm on the other hand is the natural norm induced by the L2-norm. It is defined by the square root of the maximum eigenvalue of the A^HA, where the H denotes the conjugate transpose.

$$\begin{aligned}\|A\|_2 &= \sqrt{\lambda_{max}(A^HA)}\\ &= \max_{|x|_2\neq 0}\frac{|Ax|_2}{|x|_2}\end{aligned} \tag{7.21}$$

Disappointingly, no information has been found on the convergence rate of the SPACE method, in terms of the Frobenius norm. However, comparisons can be made using the Spectral norm, and at the very least, the other three methods can be compared based off both the Frobenius and Spectral norms. There are issues associated with this as well though, because other than perhaps the CLIME and TIGER convergence rates under the Spectral norm (which were compared by Liu and Wang), comparisons are rather subjective, based on the nature of the problems in question. Another issue is that these results have been derived from different papers, so the interpretation of the results is affected. For example, for the SPACE method, especially given a scenario where n is much smaller than p, the convergence rate given suggests that it is quite possibly converges to the correct solution at a faster rate than the other methods considered, since it only relies on the number of observations. However, based off results found in other papers (Friedman et al, 2010), as well as those observed in the simulation studies carried out in this project, there is reason to believe that these convergence rates need to be examined more, which is outside the scope of this paper.

Examining the Frobenius norm results, there are some interesting aspects here to examine as well, particularly between the CLIME and TIGER methods again. Liu and Wang state that the TIGER method obtains the optimal rate of convergence under the Spectral and Frobenius norms quickly. However, depending on the nature of the problem, it appears possible that the CLIME method will converge to the solution at a faster rate than the TIGER method, simply due to the fact that the 1-norm of the precision matrix is likely to be greater than the number of non-zero off-diagonal elements in the associated matrix, as well as the (p+s) term being added to the inside of the square root for the TIGER method rate.

(7.3.2) Modification of (g)lasso

Method	Correlation	Data Matrix
Graphical Lasso	$O(p^3)$	$O(p^3)$
Symmetric group lasso	$O(p^2) + O(sp)$	$O(p^2n) + O(sn)$
Paired group lasso	$O(p^2) + O(sp)$	$O(p^2n) + O(sn)$

Table 7.2: summary of algorithm examined in the paper, with the required computational scaling

p = the number of parameters;
s = the number of non-zero elements (the same as the s in the previous section, because the main diagonal always has non-zero elements);
n = the number of observations.

As a note, in the paper by Peng et al (2009), it was stated that the SPACE method which has been examined in this paper has a complexity of $\min(O(np^2),O(p^3))$, compared with the glasso method which has a complexity of $O(p^3)$. In Section 7.1.3 the simulation study examining the convergence speeds of each of the methods has already been briefly touched upon. These results provide some more formal insight into why the symmetric group lasso and paired group lasso are considered the faster methods. However, in terms of the SPACE method, the authors state that it is somewhat mysterious that the SPACE method produces such slow results, since it was expected to perform similarly to the symmetric group lasso, however the updating of formulas in the researcher's studies could have produced a significant gain in efficiency.

(7.4) Comparing Methods

(7.4.1) TIGER vs. CLIME vs. scaled lasso vs. SCIO

There have been comparisons made between TIGER and the other methods implemented in this thesis in a few different manners. Now it may be interesting to compare this approach with a few other approaches that, even though they were not used in the simulation studies in this thesis, may provide some different perspectives on finding the sparse structure of a graphical model.

Sun and Zhang (2012) propose a method called the scaled lasso, which is a procedure that estimates each column of the precision matrix via the Scaled lasso estimator, and then adjusts the matrix estimator to be symmetric. It is stated that this procedure does not use cross-validation, instead finding the penalty level for each column via convex minimization. The advantage over methods like the glasso and CLIME methods is that because this is a column-by-column regression problem, the penalty level is automatically set to achieve the optimal rate of convergence on the regression model when estimating the corresponding column of the inverse covariance matrix. This suggests that it outperforms the glasso and CLIME methods.

Referring back to the methods section, where column-by-column regression is introduced, Sun and Zhang aim to estimate both α_j and σ_j by solving:

$$\hat{b}_j, \hat{\sigma}_j = arg \min_{b=(b_1,\ldots,b_p)^T} \left\{ \frac{b^T \hat{\Sigma} b}{2n} + \frac{\sigma}{2} + \lambda \sum_{i=1}^{p} \hat{\Sigma}_{ii} |b_i| \ \ subject\ to\ b_j = -1 \right\} \tag{7.22}$$

where, once $\hat{b}_j$ is obtained, $\alpha_j = \hat{b}_{\setminus j}$.

The next method considered was one by Liu and Luo (2012), using a Sparse Column Inverse Operator (SCIO). This approach uses some of the methodology from the CLIME method, including the same constraint for each column-wise regression, and even the same symmetrization procedure. One of the main improvements for this method is that many of the penalization methods rely on using appropriate tuning parameters based on theories derived from asymptotic or finite-samples. In this case, Cross-validation is the method used.

The SCIO estimator is defined as follows. Let $\hat{\beta}_i$ be the solution to the equation:

$$\hat{\beta}_i = arg \min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{2} \beta^T \hat{\Sigma} \beta - e_i^T \beta + \lambda_{ni} |\beta|_1 \right\}, \tag{7.23}$$

where $\beta = (\beta_1,\ldots,\beta_p)^T$ is the set of column vectors, e_i is a standard unit vector in $\mathbb{R}^p$, and λ_{ni} is the column-specific tuning parameter. Like the CLIME estimator, this approach may require a symmetrization step, as described for the CLIME approach in the Methods section.

The SCIO for the j^{th} column of the precision matrix Ω can then be solved by:

$$\hat{\Omega}_i = \arg\min_{\Omega_i} \left\{ \frac{1}{2} \Omega_i^T \hat{\Sigma} \Omega_i - e_i^T \Omega_i + \lambda_i \|\Omega_i\|_1 \right\} \tag{7.24}$$

The Ω_i refers to the i^{th} column of Ω. Technically, since the CLIME estimator implements the same column-by-column approach to estimating the precision matrix Ω, an adjustment should be made to the definition of the CLIME estimator defined in the methods section. It should, in fact, be written as:

$$\hat{\Omega}_i = \arg\min_{\Omega_i} \|\Omega_i\|_1 \ subject\ to \left\| \hat{\Sigma} \Omega_i - e_i \right\|_\infty \leqslant \delta_i \tag{7.25}$$

where this time, δ_i is the tuning parameter.

(7.4.2) Estimating the covariance matrix vs precision matrix

A small investigation will now be done on the difference between estimating the covariance matrix and the precision matrix in the context of graphical modelling. Covariance estimation in graphical modelling corresponds to find the marginal independencies between variables in a model, where a zero in the covariance matrix signifies a marginal independence between two variables. The difference between conditional and marginal independence is that in terms of marginal independence, it similar shows the dependence between two variables, disregarding a third variable, whereas for conditional independence, this third variable is considered, i.e.

$$A \perp B \qquad \text{vs} \qquad A \perp B|C$$

Bien & Tibshirani (2007) provide a sparse estimation method for the covariance matrix. In fact, it is stated that this method does for covariance matrices what the graphical lasso method does for inverse covariance matrices.

While none of the methods implemented in this thesis have focussed on estimating the sparse structure for the covariance matrix, it may be a useful path to follow for research, since it is a lesser known approach to the CIG approach.

(7.4.3) Covariance/precision matrix estimating vs graphical model estimation

Another small deviation away is to question the motivation of this research. A paper by Yuan in 2010 examined the estimation of a high dimensional inverse covariance matrix using linear programming. However, instead of using these estimations to estimate a Gaussian graphical model, the aim in this case is to focus on purely estimating these precision matrices. The reason behind this is that the ability to approximate the graphical model with a relatively low degree determines how well the estimation method finds the high dimensional precision matrix. Therefore, focusing on estimating the inverse covariance matrix means that it can identify the sparse structure of the matrix better than the methods which have the aim of providing a graphical model.

The author goes on to compare the approach proposed in this paper, with the neighbourhood pursuit considered by Meinhausen and Bühlmann in 2006. It is revealed that when the target matrices are sparse or approximately sparse, then the estimation method (Yuan's approach) and the selection approach (Meinhausen and Bühlmann) provide different results. Another advantage of the estimation method is that there are

weaker assumptions required to carry out analysis, compared to the neighbourhood pursuit approach. Insight is even provided on issues behind the glasso method, where it is found, through a theorem, that the glasso method penalizes the maximum likelihood estimate based on the total number of edges, whereas the penalty should be based off the degree of the model itself.

Again, the motivation of this thesis is to provide a graphical representation of the conditional independencies between variables, so it does not seem suitable to use an estimation method such as that proposed by Yuan. But it at the very least provides some insight into the caveats occurring in some of the methods implemented in this research.

(8) Future Research

While the results produced in this research has provided many opportunites for discussion, there still exists the necessity to consider future research. One major point of consideration is that of the simulation study. Section 8.1 considers alternatives and improvements on the analysis conducted in this research. Section 8.2 develops on this point, allowing for the use of different types of approaches in future analysis. Finally section 8.3 looks at the different methods that could be used in the studies, and some modifications that could be made to the approaches considered in this thesis.

(8.1) Different Simulations

The simulation studies in this approach were derived from previous research in the area of graphical modelling, focusing on the performances of the SIN and GMTS methods in find the correct true model based on the simulation. This simulation was produced for models which followed a multivariate Gaussian distribution, with Gaussian error terms. The paper by Gottard and Pacillo in 2010 brought up an interesting point about the SIN method in particular. Since it uses the sample estimator of the covariance matrix in producing the p-values for the partial correlation matrices, it is sensitive to outliers in the data. This may provide some motivation to examine other types of simulations, to determine whether the SIN method would still produce favourable results, particularly over the ℓ_1-penalized approaches (although it is likely these methods would be affected too).

Developing on this notion, the ℓ_1 -regularization methods were first proposed to combat the issues associated with high-dimensional datasets, where the number of parameters, p, significantly exceeded the number of observations, n. In the simulation studies conducted in this research, the closest possible dataset to this scenario is the SVAR(3) with 1000 observations, which well and truly provides a full rank empirical matrix. In order to carry out a more detailed comparison between the original GMTS and SIN methods, and these four other models implemented, it may be useful to simulate a process where $p \gg n$, in order to determine which method performs the best in the scenario they were designed for.

(8.2) Different Distributions

This research has focused on a Multivariate Normal time series setting, due to convenience with providing the partial correlations between variables. However, it

would be helpful if it was possible to implement graphical modelling using data which follows other distributions. In fact, Liu et al (2012) state two drawbacks of implementing Gaussian graphical models. The first is that most datasets generally do not follow a Gaussian distribution, and the second is that the data could be subject to noise or outliers (similar to the issues with the SIN approach). Liu et al (2009) state that in a high-dimensional setting instead of using the graphical lasso approach like that used in the normal-parametric approach, a sparse additive model should be used, results in a ℓ_1-regularized nonparanormal graphical model.

This leads into an entirely new area of graphical modelling. The simulations conducted in this research originated from a multivariate Gaussian distribution, which was not subject to outliers. It would be interesting to make comparisons between results from different datasets which originated form different distributions, instead of simply datasets of different sample sizes coming from the same simulation.

(8.2.1) More extensive studies into graphical lasso and its modifications

The results in this research suggested that the graphical lasso, while may be a slower process than the other, is a useful method in providing the structure of a graphical model. However, as observed in these discussions, there are issues associated with this method, including issues with positive definiteness and convergence to the solution. At the very least, it would be interesting to see whether comparing the newer methods discussed (like dp-glasso and the symmetric lasso), to the glasso would provide any interesting results.

(8.3) Different methods

Recently, there has been a resurgence in the implementation of greedy forward/backward selection procedures, similar to that described in the start of the methods section, by Edwards (2000). The motivation for this type of method lies in the fact that the full structure of the model can be learnt with a high probability with just O(d log(p)) samples, which, when compared with one of the methods used in this thesis, the glasso, is a vast improvement, since the glasso requires (d^2 log(p)) samples. In the approach proposed by Johnson et al in 2011, a combination of the forward-backward greedy algorithms was considered. The idea was to start with an empty set of variables, with the first step being to find the best next "candidate" (variables) to the active set, only if it improves the loss function used by a significant amount. The next step (the backward step) checked the influence of all the variables on the newly added one. If at

least one of the variables added before does not contribute a significant amount to the loss function, then the algorithm removes them from the active set.
While this seems like an interesting approach to compare with the other methods used in this thesis, questions still remain with the complexity of the problem, and whether this approach will be slower than the other approaches in a high-dimensional setting.

(9) Conclusions

Throughout this thesis the motivation has been to convince the reader of the usefulness of graphical modelling in the context of Multivariate Time Series. The way of showing the advantages of this type of model was to consider it when representing structural vector autoregressive models (SVAR) of varying orders. Previous research in this area, which was developed upon in this thesis, in the form of the GMTS and SIN approaches, were proven to still be very important approaches to consider when aiming to discover the structure of the graphical model.

The SVAR model has been shown to provide a useful platform to determine the dependencies between not only the contemporaneous and lagged variables, but also between solely the contemporaneous variables. Simply being able to use the coefficients of each regression to determine which variables are linearly dependent is an attractive prospect. There are also other aspects of analysis associated with SVAR models, such as impulse analysis and forecast error variance decompositions, which could be examined further in other analyses.

An issue becomes apparent when comparing with the ℓ_1-regularization methods introduced in this thesis. When the aim of the research is to provide a sparse structure for the model, the SIN and GMTS merely provide different tests for conditional independencies between variables, and do not have a method of inducing sparsity in the structure of the graphical model, like the penalization methods in ℓ_1-regularization. In this thesis however, it was determined that perhaps penalization methods are not always necessary when it comes to finding the Conditional Independence Graph (CIG) of the dataset, as shown by the superior results from the GMTS and SIN methods. It must be noted however that the ℓ_1-regularization methods do require further analysis, because it is still unknown how these approaches could provide relatively poor results.

Even in terms of the convergence rates, the simpler testing procedures (GMTS and SIN) were a more attractive prospect using these simulated datasets, due to the lack of optimization required in the process. A more formal comparison of the speeds of convergence is required for each of the ℓ_1-regularization methods, since basic observations of process speeds does not provide any proper results in the context of research in this area.

It has been discovered that while the main motivation behind using the ℓ_1-regularization remains (approximately) true for each of the four methods examined in this thesis, there are many differences, great or small, between them. This can be seen in the function that requires penalization in each method, the manner in which the tuning parameter is selected in each of the methods, or even the penalty itself, which is imposed on the function, which in this case can be based on either the precision matrix or the partial correlation matrix.

In Section 8 of this thesis, there were many areas of future research considered, that were outside the motivations of this thesis. It would be useful to examine the performances of these six methods using different simulation studies. In particular, due to the knowledge that the ℓ_1-regularization methods are designed to cope with high-dimensional problems, $p \gg n$, a set of simulation studies, similar to the one conducted in this thesis, could be carried out to determine whether the GMTS and SIN methods would outperform the ℓ_1-regularization methods again. If it is found that the ℓ_1-regularization methods do perform better, then perhaps it would be useful to consider a more comprehensive analysis of these approaches, considering some of the methods not used in this research (like the SCIO by Liu and Luo (2012), and the scaled-lasso by Sun and Zhang (2012).

(10) Bibliography

[1] Banerjee, O., El Ghaoui, L., & d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *The Journal of Machine Learning Research*, *9*, 485-516.

[2] Banerjee, O., Ghaoui, L. E., d'Aspremont, A., & Natsoulis, G. (2006, June). Convex optimization techniques for fitting sparse Gaussian graphical models. In *Proceedings of the 23rd international conference on Machine learning* (pp. 89-96). ACM.

[3] Belloni, A., Chernozhukov, V., & Wang, L. (2011). Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika*, *98*(4), 791-806.

[4] Bien, J., & Tibshirani, R. J. (2011). Sparse estimation of a covariance matrix. *Biometrika*, *98*(4), 807-820.

[5] Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

[6] Brüggemann, R. (2004). *Model reduction methods for vector autoregressive processes* (Vol. 536). Springer Science & Business Media.

[7] Bühlmann, P., & Van De Geer, S. (2011). *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media.

[8] Cai, T. T., Liu, W., & Zhou, H. H. (2012). Estimating sparse precision matrix: Optimal rates of convergence and adaptive estimation. *arXiv preprint arXiv:1212.2882*.

[9] Cai, T. T., Zhang, C. H., & Zhou, H. H. (2010). Optimal rates of convergence for covariance matrix estimation. *The Annals of Statistics*, *38*(4), 2118-2144.

[10] Cai, T., Liu, W., & Luo, X. (2011). A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, *106*(494), 594-607.

[11] d'Aspremont, A., Banerjee, O., & El Ghaoui, L. (2008). First-order methods for sparse covariance selection. *SIAM Journal on Matrix Analysis and Applications*, *30*(1), 56-66.

[12] Dempster, A. P. (1972). Covariance selection. *Biometrics*, 157-175.

[13] Drton, M., & Perlman, M. D. (2007). Multiple testing and error control in Gaussian graphical model selection. *Statistical Science*, 430-449.

[14] Drton, M., & Perlman, M. D. (2008). A SINful approach to Gaussian graphical model selection. *Journal of Statistical Planning and Inference*, *138*(4), 1179-1200.

[15] Edwards, D. (2000). *Introduction to graphical modelling*. Springer Texts in Statistics.

[16] Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1). Springer, Berlin: Springer series in statistics.

[17] Friedman, J., Hastie, T., & Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, *9*(3), 432-441.

[18] Friedman, J., Hastie, T., & Tibshirani, R. (2010). *Applications of the lasso and grouped lasso to the estimation of sparse graphical models* (pp. 1-22). Technical report, Stanford University.

[19] Gottard, A., & Pacillo, S. (2010). Robust concentration graph model selection. *Computational Statistics & Data Analysis*, *54*(12), 3070-3079.

[20] Hsieh, C. J., Dhillon, I. S., Ravikumar, P. K., & Sustik, M. A. (2011). Sparse inverse covariance matrix estimation using quadratic approximation. In *Advances in Neural Information Processing Systems* (pp. 2330-2338).

[21] James, G. M., Radchenko, P., & Lv, J. (2009). DASSO: connections between the Dantzig selector and lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *71*(1), 127-142.

[22] Johnson, C. C., Jalali, A., & Ravikumar, P. (2011). High-dimensional sparse inverse covariance estimation using greedy methods. *arXiv preprint arXiv:1112.6411*.

[23] Li, X., Zhao, T., Yuan, X., & Liu, H. (2012). An R Package flare for High Dimensional Linear Regression and Precision Matrix Estimation. *R Package Vignette*.

[24] Lin, A., (2008). Edge Deletion Tests in Graphical Modelling for Multivariate Time Series. *Honours Project dissertation (unpublished), University of Canterbury.*

[25] Liu, H., & Wang, L. (2012). Tiger: A tuning-insensitive approach for optimally estimating gaussian graphical models. *arXiv preprint arXiv:1209.2437*.

[26] Liu, H., Han, F., Yuan, M., Lafferty, J., & Wasserman, L. (2012). High-dimensional semiparametric Gaussian copula graphical models. *The Annals of Statistics*, *40*(4), 2293-2326.

[27] Liu, H., Lafferty, J., & Wasserman, L. (2009). The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *The Journal of Machine Learning Research*, *10*, 2295-2328.

[28] Liu, W., & Luo, X. (2012). High-dimensional sparse precision matrix estimation via sparse column inverse operator. *arXiv preprint arXiv:1203.3896*.

[29] Lütkepohl, H. (2005). *New introduction to multiple time series analysis*. Springer Science & Business Media, New York.

[30] Mazumder, R., & Hastie, T. (2012). The graphical lasso: New insights and alternatives. *Electronic journal of statistics*, *6*, 2125-2149.

[31] Meinshausen, N., & Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 1436-1462.

[32] Peng, J., Wang, P., Zhou, N., & Zhu, J. (2009). Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association*, *104*(486).

[33] Pfaff, B. (2008). VAR, SVAR and SVEC models: Implementation within R package vars. *Journal of Statistical Software*, *27*(4), 1-32.

[34] Sims, C. A. (1980). Macroeconomics and reality. *Econometrica: Journal of the Econometric Society*, 1-48.

[35] Sun, T., & Zhang, C. H. (2013). Sparse matrix inversion with scaled lasso. *The Journal of Machine Learning Research*, *14*(1), 3385-3418.

[36] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267-288.

[37] Tsay, R. S. (2013). *Multivariate Time Series Analysis: With R and Financial Applications*. John Wiley & Sons.

[38] Verzelen, N., & Villers, F. (2009). Tests for Gaussian graphical models. *Computational Statistics & Data Analysis*, *53*(5), 1894-1905.

[39] Vichik, S., & Oshman, Y. (2011, June). Optimal covariance selection for estimation using graphical models. In *American Control Conference (ACC), 2011* (pp. 5049-5054). IEEE.

[40] Wilson, G. T. (2010). *Atmospheric CO2 and global temperatures: the strength and nature of their dependence*. Working paper.

[41] Wilson, G. T., Reale, M., & Morton, A. S. (2001). *Developments in multivariate time series modeling*. Department of Mathematics and Statistics, University of Canterbury.

[42] Wilson, G. T., & Reale, M. (2008). The sampling properties of conditional independence graphs for I (1) structural VAR models. *Journal of Time Series Analysis*, *29*(5), 802-810.

[43] Xue, L., & Zou, H. (2012). Regularized rank-based estimation of high-dimensional nonparanormal graphical models. *The Annals of Statistics*,*40*(5), 2541-2571.

[44] Yuan, M. (2010). High dimensional inverse covariance matrix estimation via linear programming. *The Journal of Machine Learning Research*, *11*, 2261-2286.

[45] Yuan, M., & Lin, Y. (2007). Model selection and estimation in the Gaussian graphical model. *Biometrika*, *94*(1), 19-35.

Appendix A

Simulating Multivariate Time Series

The simulation studies examined in this research, as alluded to previously, borrows from previous work in the area, which focussed on comparing the GMTS and SIN methods. It is still useful to describe how these datasets were simulated.

This method of simulation was originally introduced in Lütkepohl (2006). The steps to simulating this data are as follows:

Step 1: Generating the n error terms

Firstly generate the error terms in the model using the univariate standard normal distribution:

$$\epsilon_t' = \begin{pmatrix} \epsilon_{1,1} & \epsilon_{1,2} & \dots & \epsilon_{1,N} \\ \epsilon_{2,1} & \epsilon_{2,2} & \dots & \epsilon_{2,N} \\ \epsilon_{3,1} & \epsilon_{3,2} & \dots & \epsilon_{3,N} \end{pmatrix}$$

where $\epsilon_{i,j} \sim N(0,1)$, i = 1, 2, 3, j = 1, 2, …, N

Next $\epsilon_t^{'}$ must be multiplied by the Cholesky decomposition of $\Sigma\epsilon$, where

$$\Sigma_\epsilon = \begin{pmatrix} 0.09 & 0 & 0 \\ 0.09 & 0.4 & 0 \\ 0 - 0.09 & 0 & 0.025 \end{pmatrix}$$

$$\epsilon_t = P\epsilon_t'$$

where PP' = Σ_ϵ

Step 2: Obtaining multiple time series X_t

In order to obtain the multivariate time series $X_t = (x_{1,t}, x_{2,t}, x_{3,t})$, initial values are required at the start of the simulatons. Lütkepohl (2006) deduced that the covariance matrix of X_t should be used, and this ensured that the same correlation structure for the initial values, as well as the rest of the time series, would exist.
This covariance matrix Σ_X is obtained by:

$$Vec(\Sigma_X) = \left(I_{kp}^2 - \Phi \otimes \Phi\right)^{-1} Vec(\Sigma_\epsilon)$$
$$= (I_{3x2}^2 - \Phi \otimes \Phi)^{-1} Vec(\Sigma_\epsilon)$$

where

$$\Phi = \begin{pmatrix} \Phi_1 & \Phi_2 \\ I_3 & 0_3 \end{pmatrix} = (I_{3x2}^2 - \Phi \otimes \Phi)^{-1} Vec(V)$$

where Φ_1 amd Φ_2 are two coefficient matrices from the equation:

$$\begin{pmatrix} x_t \\ y_t \\ z_t \end{pmatrix} \begin{pmatrix} 0.9 & 0 & 0 \\ 0.9 & 0.6 & 0 \\ -0.9 & 0 & -0.5 \end{pmatrix} \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ z_{t-1} \end{pmatrix} \begin{pmatrix} -0.6 & 0 & 0 \\ 0.3 & 0 & 0.4 \\ 0.6 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_{t-2} \\ y_{t-2} \\ z_{t-2} \end{pmatrix}$$

$\Phi \otimes \Phi$ is the Kronecker-product, as illustrated below:

Suppose that

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \qquad B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$$

Then

$$A \otimes \mathrm{B} = \begin{pmatrix} a_{11}B & a_{12}B \\ a_{21}B & a_{22}B \end{pmatrix} = \begin{pmatrix} a_{11}b_{11} & a_{11}b_{12} & a_{12}b_{11} & a_{12}b_{12} \\ a_{11}b_{21} & a_{11}b_{22} & a_{12}b_{21} & a_{12}b_{22} \\ a_{21}b_{11} & a_{21}b_{12} & a_{22}b_{11} & a_{22}b_{12} \\ a_{21}b_{21} & a_{21}b_{22} & a_{22}b_{21} & a_{22}b_{22} \end{pmatrix}$$

Taking Cholesky decompositions of Σ_X: QQ' = Σ_X, the starting values of X_t are found by:

$$\begin{pmatrix} x_{1,1} \\ x_{2,1} \\ x_{3,1} \\ x_{1,2} \\ x_{2,2} \\ x_{3,2} \end{pmatrix} = Q \begin{pmatrix} \epsilon_{1,1} \\ \epsilon_{2,1} \\ \epsilon_{3,1} \\ \epsilon_{1,2} \\ \epsilon_{2,2} \\ \epsilon_{3,2} \end{pmatrix}$$

These then become the values of X_{t-2} and X_{t-1} respectively.

Step 3: Simulating the time series vectors.

The equations for simulating the data for the VAR(2) are given as:

$$x_{1,t} = \quad 0.9x_{1,t-1} - 0.6x_{1,t-2} + \epsilon_{1,t}$$
$$x_{2,t} = x_{1,t} + 0.6x_{2,t-1} + 0.9x_{1,t-2} + 0.4x_{3,t-2} + \epsilon_{2,t}$$
$$x_{3,t} = \quad -x_{1,t} - 0.5x_{3,t-1} + \epsilon_{3,t}$$

In order to obtain a sample size of n in the dataset, a sequence of size N observations is simulated, with $N \gg n$, then the first $N - n$ values are treated as a burn-in period, and discarded. In the case of this research, all simulations had a burn-in number of 500.